DEPARTMENT OF TRANSPORTATION

Generating Traffic Information from Connected Vehicle V2V Basic Safety Messages

Michael Levin, Principal Investigator
Department of Civil, Environmental, and Geo- Engineering
University of Minnesota

MARCH 2021

Research Project
Final Report 2021-08

To request this document in an alternative format, such as braille or large print, call 651-366-4718 or 1-800-657-3774 (Greater Minnesota) or email your request to ADArequest.dot@state.mn.us. Please request at least one week in advance.

Technical Report Documentation Page

1. Report No.	2.	3. Recipients Accession No.	
MN 2021-08			
4. Title and Subtitle Generating Traffic Information from Connected Vehicle V2V Basic Safety Messages		5. Report Date March 2021 6.	
7. Author(s) Rongsheng Chen, Michael Levin, John Hourdos, Melissa Duhn		8. Performing Organization Report No.	
9. Performing Organization Name and Address Department of Civil, Environmental, and Geo- Engineering University of Minnesota 500 Pillsbury Drive S.E. Minneapolis, MN 55455		10. Project/Task/Work Unit No. CTS#2019020 11. Contract (C) or Grant (G) No. (C) 1003325 (WO) 65	
12. Sponsoring Organization Name and Address Minnesota Department of Transportation Research Services & Library 395 John Ireland Boulevard, MS 330 St. Paul, Minnesota 55155-1899		13. Type of Report and Period Covered Final Report 14. Sponsoring Agency Code	
15. Supplementary Notes https://mndot.gov/research/reports/2021/202108.pdf			

16. Abstract (Limit: 250 words)

Basic Safety Message (BSM) containing data about the vehicle's position, speed, and acceleration. Roadside receivers, RSUs, can capture BSM broadcasts and translate them into information about traffic conditions. If every vehicle is equipped with awareness, BSMs can be combined to calculate traffic flows, speeds, and densities. These three key parameters will be post-processed to obtain queue lengths and travel time estimates. The project team proposed a traffic state estimation algorithm using BSMs based on the Kalman filter technique. The algorithm's performance was tested with BSMs generated from several arterial in a microscopic simulation model and BSMs generated with radar data collected on freeway sections. Then the project team developed a traffic monitoring system to apply the algorithm to a large-scale network with different types of roads. In the system, computers could remotely access the online server to acquire BSMs and estimate traffic states in real-time.

17. Document Analysis/Descriptors	18. Availability Statement		
Vehicle to vehicle communications, traffic estimation, connected vehicles	No restrictions. Document available from: National Technical Information Services, Alexandria, Virginia 22312		
19. Security Class (this report) Unclassified	20. Security Class (this page) Unclassified	21. No. of Pages 177	22. Price

Generating Traffic Information from Connected Vehicle V2V Basic Safety Messages

FINAL REPORT

Prepared by:
Rongsheng Chen
Michael W. Levin
John Hourdos
Melissa Duhn
Department of Civil, Environmental, and Geo- Engineering
University of Minnesota

March 2021

Published by:
Minnesota Department of Transportation
Research Services & Library
395 John Ireland Boulevard, MS 330
St. Paul, Minnesota 55155-1899

Acknowledgements

This project was funded by the Minnesota Department of Transportation.

List of Abbreviations

BSM – Basic Safety Message (SAE J2735)
BSMMS –Basic Safety Message Monitoring System
CA – Certificate Authority
CV – Connected Vehicle
C-V2X – Cellular based Vehicle to Anything communication
DOT – Department of Transportation
DSRC – Dedicated Short-Range Communication (IEEE 802.11p and IEEE 1609)
FHWA – Federal Highway Administration
GPS – Global Positioning System. The USA version of Global Navigation Satellite System (GNSS)
IP – Internet Protocol
ITS – Intelligent Transportation Systems
MnDOT – Minnesota Department of Transportation
OBU – On-Board Unit
PKI – Public Key Infrastructure
PoE – Power over Ethernet
PSM – Pedestrian Safety Message
RSE – Road-Side Equipment
RSU – Road-Side Unit
RTCM – Radio Technical Commission for Maritime Services Message (SAE J2735)
SAE – Society of Automotive Engineers
SCMS – Security Credential Management System
SDK – Software Development Kit
SPaT – Signal Phasing and Timing (SAE J2735)
SRM – Signal Request Message (SAE J2735)
SSM – Signal Status Message (SAE J2735)
TIM – Traveler Information Message (SAE J2735)
TMC – Traffic Management Center
TSR –Traffic Sign Recognition
USDOT – United States Department of Transportation
V2I – Vehicle-to-Infrastructure (communication)
V2V – Vehicle-to-Vehicle (communication)
V2X – Vehicle-to-Everything (e.g. Infrastructure, Vehicle, etc.) (communication)

Contents

Executive Summary

Background and Approach

Traffic information plays a vital role in the operation of intelligent traffic systems. Traditional traffic detectors, such as loop detectors and video detectors, are installed at fixed locations and need a high maintenance fee. The development of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) technologies enable vehicles to share their travel messages with other vehicles or the roadside infrastructure with a high frequency. The most frequent and basic group of data broadcasts is called the Basic Safety Messages (BSMs). BSMs were originally proposed for some safety applications to reduce accident rates on the road, such as alert drivers about the occurrence of extreme events. In this project, BSMs are used as the data source for traffic state estimation (including density, speed, and flow), which could replace traditional detectors on the road.

In this project, the research team proposes a real-time traffic state algorithm using BSMs and tests its accuracy with BSMs generated from microscopic traffic simulation and radar data. The research team mainly completes the following tasks:

- Task 1: Data collection and emulated BSM generation. The research team prepares two types of BSM data. One type of BSMs is generated by a calibrated microscopic traffic simulation model, and the second type of BSMs is generated from radar data collected in the field
- Task 2: Develop BSM Traffic State Estimator using mesoscopic flow model. The research team creates a traffic state estimation model based on the Kalman filter (KF) technique and the cell transmission model (CTM).
- Task 3: Accuracy analysis. The research team tests the accuracy of the developed algorithm using the data collected in Task 1.
- Task 4: Improve TSE accuracy using existing sensor data. The research team explores how much the sensor data at link entry can improve the algorithm accuracy.
- Task 5: Develop functional specifications for network-wide BSM-based traffic monitoring. The research team develops the functional specification for the future traffic monitoring system.
- Task 6: Development of BSM Traffic Monitoring system prototype. The research team develops a functional prototype of the BSM traffic monitoring system. The traffic monitoring

system is composed of data collection, data transmission, and data processing modules, which is able to upload BSMs to servers, download BSMs from servers, and conduct real-time traffic estimation.

Key Findings

There are some findings regarding the performance of the algorithm.

- The accuracy of speed estimation is much higher than the density estimation and flow estimation.
- The accuracy of density estimation and flow estimation cannot outperform traditional loop detectors.
- If the algorithm is applied to a large network, enough computing power needs to be prepared to guarantee the real-time operation of the traffic monitoring system.

1 Introduction

Traffic information plays an important role in the operation of an intelligent traffic system. Traffic models need adequate information to calibrate their parameters so that the traffic management and control system can react to the variation in traffic conditions and maintain the efficiency of traffic networks. Traffic information is collected by traffic detectors, such as loop detectors and video detectors. These detectors are installed at specific locations and are hard to maintain. For example, the loop detector only collects data at specific locations and it needs to be embedded in the pavement. The video detector should be installed at high buildings. Its detection is greatly affected by the weather and light condition, and the processing of video image data requires a lot of time and storage. The accuracy of video data is also limited by the location of the cameras.

The Connected Vehicle Safety Pilot Program initialized by the U.S Department of Transportation (USDOT) aims to test safety applications that use dedicated short-range communications (DSRC) in real-world scenarios, which include vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) technologies. In this program, the Safety Pilot Model Deployment used about 3,000 vehicles equipped with wireless devices to create a test field with V2V and V2I communications, with which a vehicle can share its travel messages with other vehicles or the road-side infrastructure with a high frequency, and the most often and basic group of data broadcast is called the Basic Safety Messages (BSMs). The major information included in BSMs is shown in Table 1. The main data relevant to this paper are the position and velocity of each vehicle. Temporal IDs of vehicles are updated periodically to preserve personal privacy. The position of the vehicle is identified by the Global Positioning System (GPS) set of coordinates.

BSMs were originally proposed for some safety applications to reduce the accident rate on the road. The first application is to alert drivers about the occurrence of extreme events. Helping drivers to notice dangerous driving environments helps them react to extreme events in advance and can effectively avoid traffic accidents. Liu (25) proposed a methodology to detect high acceleration, and Khattak et al. (21) developed a model to distinguish normal or anomalous driving behaviors, which was important for hazard anticipation and notification systems. The second application is a collision warning system. A forward collision warning system detects the speed of the preceding vehicle and the distance to the preceding vehicle. The driver of the rear vehicle will be warned if the current speed is too high or the vehicle is too close to the preceding one. With V2V communication, a cooperative collision warning system can provide warnings or awareness displays for drivers based on the driving states of vehicles that are either in sight of or out of the sight of the current vehicle (33). The third application is a red light violation warning. When the red light violation warning system at an intersection detects that a vehicle approaching the intersec-

tion may potentially violate the red light, the system will send warnings to the in-vehicle device to notice the driver in the vehicle. The cooperative intersection signal violation warning system was proven in studies to reduce red light violations and intersection collisions using in-vehicle warning devices (30). There are other safety applications based on BSMs that are developed and tested to reduce the crash rate at urban roads or freeways, such as lane change warning and work zone warning.

Table 1: Major vehicle travel information included in BSMs

index	information
1	Temporal ID
2	Timestamp
3	Position
4	Speed
5	heading
6	Acceleration
7	Brake system status
8	Vehicle size
9	SteeringWheel Angle
10	Positional Accuracy

Most traffic safety applications with BSMs focus on assisting each driver individually. These applications help drivers become aware of the potential danger and make them take measures to prevent traffic accidents. Unlike these applications, traffic controls applied to an entire road section, such as ramp metering or variable speed limits, control traffic flows instead of individual vehicles. These traffic controls require knowledge of the traffic state because the actual traffic condition determines the control applied to the road. Traditional traffic detectors, such as loop detectors and video cameras, are widely used for collecting data for traffic state estimation, but they have some inevitable drawbacks. Loop detectors installed in pavement are frequently broken and need regular maintenance. Camera detectors have low accuracy in adverse weather conditions. Neither detector can cover the entire road section because they are installed at fixed locations. The measurement accuracy decreases at large distances from detectors. Information from BSMs provides us the probability of estimation traffic conditions for the entire road without the limitations of location and installation of traffic detectors. BSM data is similar to the probe vehicle data, which is collected by running vehicles equipped with data collection facilities. Some probe vehi-

cles are installed with video cameras that enable them to measure the spacing between vehicles. However, it is costly for probe vehicles to reach a high penetration rate. Probe vehicles require drivers and are not freely available in normal traffic. Unlike probe vehicles, vehicles with V2V or V2I communication modules can reach a high market penetration as they are supported by the Department of Transportation. As the market penetration of connected vehicles becomes larger, BSM data can cover a large area, and traffic state estimation using BSMs will have a high accuracy.

In this project, the project team proposed a method for traffic state estimation using BSMs based on the Kalman filter technique. The estimator computed traffic measurements (density, speed, and flow) on road sections without the need of loop detectors. At first, the estimator was tested with BSMs generated from a network built using the cell transmission model (CTM). Then BSMs generated from two freeway sections in a microscopic simulation model in traffic simulation package AIMSUN and from radar data collected in the field were used to test the accuracy of the estimator. To explore whether the estimation accuracy can be improved given accuracy input flow measurements, the estimator was tested when the input flow was provided. After that, the project team explored the capability of this algorithm applying to a large-scale network and developed a traffic monitoring system that is able to conduct traffic state estimation using BSMs stored in a remote server.

The structure of this report is as follows: Chapter 2 summarizes existing studies related to traffic state estimation. Chapter 3 introduces the methodology used for traffic state estimation in this study. Chapter 4 shows the experiment using the CTM-based simulator. Chapter 5 illustrates the process of preparing databases with BSMs generated from a microscopic simulation model. Chapter 6 shows the experiment on two freeway sections in AIMSUN. Chapter 7 presents the work for preparing BSMs generated from radar data collected on freeways. Chapter 8 shows the experiment using BSMs generated from radar data collected from a short freeway section in the field. Chapter 9 shows the experiment when the measurement of input flow is given. Chapter 10 illustrates the structure of the traffic monitoring system and its performance on a large scale network in AIMSUN.

2 Literature review

Traffic state estimation is the process of calculating actual values of traffic state variables given traffic data with errors or data that only covers part of vehicles in the network. The traffic state variables include flow, density, and speed.

The data source for traffic state estimation evolves with the development of communication technologies in traffic systems. There are studies using loop detector data (43; 39; 42; 10), probe vehicle data (27; 46; 13; 35), and multiple data resources from both loop detectors and probe vehicles (22; 9; 26). Probe vehicle data differs among studies. In some studies, probe vehicles are only equipped with GPS devices to record the geo-position (46) of vehicles while probe vehicles in other studies are also equipped with cameras and are capable of measuring distances to preceding vehicles (35).

Most studies using the probe vehicle data focus on estimating only one traffic state variable, such as speed (45), density (17; 35), or travel time (27; 16; 12; 48), while studies using loop detector data usually estimate multiple traffic state variables (44; 42). Work et al. (45) derived a velocity model based PDE on the Lighthill-Whitham-Richard PDE. A velocity cell transmission model (CTM-v) was built after integrating the new PDE with the derived velocity model. The Ensemble Kalman filtering (EnKF) technique was used to estimate the velocity based on the GPS data. The velocity estimation was validated with microsimulation data and historical velocity data and proven to have less error than the averaging scheme. Hellinga et al. (16) estimated average link travel times with low-frequency anonymous probe car data. This travel time allocation problem was solved by a travel time decomposition method which was able to compute the congestion time and the stopping time on a link. The result showed that this method improved the accuracy of travel time estimation by 40% on average with a frequency of one-minute compared with the baseline method, which proportionally assigned the travel time on each link according to its free flow travel time. Herring et al. (18) utilized taxi trajectory data to estimate arterial traffic conditions. A fleet of 500 taxis was used to collect GPS data with a frequency of one minute. An expectation maximization algorithm was developed to estimate parameters in a Coupled Hidden Markov Model (CHMM). Then CHMM was used for the prediction of link travel time. The result showed that this model had a higher accuracy than the baseline model by 35%. Zhu et al. (48) proposed an algorithm to predict the travel times for urban arterial roads based on Kalman filtering using the probe car data. The hierarchical clustering was used to estimate the parameters in the model. The results showed that the proposed algorithm could be applied to provide real-time travel time prediction services. Most probe vehicles cannot directly obtain traffic density and flow measurements on the road because probe vehicles cannot cover 100% vehicles on the road, which makes it difficult for

studies with probe vehicles to estimate traffic densities and traffic flows than studies using loop detector data.

Based on the methodology used for traffic state estimation, existing studies can be categorized into model-driven and data-driven studies. Most model-based studies estimate traffic states based on calibrated traffic flow models which provide macroscopic relationships between traffic state variables and capture traffic flow dynamics on the road. Some studies use first-order traffic flow models, in which the flow-density relationship can be illustrated by different traffic fundamental diagrams, such as triangular fundamental diagram (40; 35), trapezoidal fundamental diagram (36), and Greenshields flow function (45). Other studies use second-order traffic flow models, such as METANET (43). Wang et al. (43) proposed a general approach to estimate the traffic state of a freeway in real-time based on an extended Kalman filter. A macroscopic traffic flow model was used as the prediction step in the Kalman filter while loop detector data supported the measurement step. Several simulations were conducted to test the model accuracy and the effect of parameter estimation on traffic state estimation. The study emphasized the importance of parameter estimation for accurate traffic state estimation and the model was sensitive to the initial values of the model parameters. The results showed that a well-designed traffic state estimator along with real-time model parameter estimation can produce estimation with high accuracy. A few studies proposed parameter estimation methods (34; 41) while other studies directly used traffic flow models calibrated using other data source (27).

A data assimilation technique called Kalman filter (19) is widely used in existing model-driven studies for traffic state estimation. It is a recursive algorithm that integrates measurements and model-based predictions. Both of them contain errors and other uncertainties. Kalman filters are able to generate an estimation that is more accurate than both individual measurement and prediction values. There are different types of filters used for traffic state estimation, such as extended Kalman filter (40; 44), ensemble Kalman filter (45), and particle Kalman filter (20). They differ in their assumptions about the system. For example, whether the system dynamic is linear or whether the model function is derivable.

Unlike model-driven studies, for most data-driven studies of traffic state estimation, traffic flow models are not needed. The relationship between the target traffic state and the available data source are captured by statistical models based on historical data. Liang et al. (23) created a deep generative adversarial architecture (GAA) for network-wide spatial-temporal traffic-state estimation. GAA mainly used an artificial neural network (ANN) to capture the correlation between the input and the output but the relationship between traffic state variables cannot be directly derived from this model. Other models used in data-driven study for traffic state estimation include a Coupled Hidden Markov Model (CHMM) (18) and a multi-layer ANN (47).

In this project, we use a similar methodology to previous model-driven approaches. CTM, a first-order traffic flow model, is used to represent traffic dynamics. Kalman filter technique is used to estimate the three traffic state variables of traffic flow, density, and speed. Unlike previous studies, BSM data is the only data source used for estimation in this study, no detector data is used. Besides, existing studies that estimate traffic density often have measurements of density, but this study only has measurements of speed.

3 Methodology

In this section, a Kalman filter is used to estimate traffic states. Kalman filtering (19) is an algorithm that integrates measurements and model-based predictions. Both of them contain errors and other uncertainties. The Kalman filter is able to generate an estimation that is more accurate than any of these single measurements or predictions. The Kalman filter iterates between the measurement step and the prediction step.

3.1 Cell Transmission model

In the prediction part, the cell transmission model is applied to model the dynamics of traffic flow. CTM is a discretization of the Lighthill-Whitham-Richards partial differential equation (24; 31) using a Godunov scheme. The parameters in CTM are based on a fundamental diagram, such as triangular or trapezoidal diagrams. Figure 1 shows a triangular fundamental diagram of traffic flow q as a function of density k. It includes four parameters: the capacity Q, the free-flow speed v_f, the jam density K, and the backward shockwave speed w. In the cell transmission model with a time step size of Δt, the road section was divided into several cells with cell length $\Delta L = \Delta t \cdot v_f$. The cell density for each cell can be updated using equation (1). The cell density $n_i(t)$ mentioned in this report is not the percentage of time when a loop detector is occupied but represents the number of vehicles in a cell i at a time step t, which equals the product of density and the cell length. In equation (1), the density n of cell i at time $t+1$ equals the density of cell i at time t minus the exiting flow from the current cell, y_i, plus the entering flow from the previous cell, y_{i-1}.

$$n_i(t+1) = n_i(t) - y_i(t) + y_{i-1}(t) \tag{1}$$

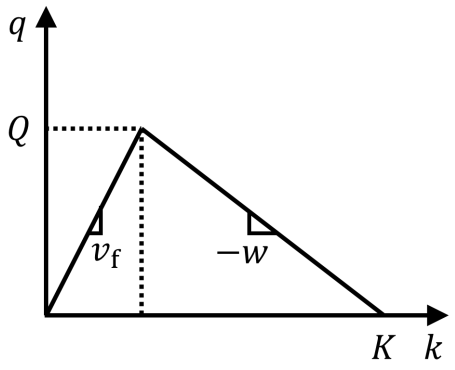


Figure. 1: Trapezoidal fundamental diagram

Equation (2) shows how the inter-cell flow is calculated.

$$y_i(t) = \min\{n_i(t), Q\Delta t, \frac{w}{v_f}(N_{i+1} - n_{i+1}(t))\} \tag{2}$$

There are three terms in the minimization function of equation (2). The first term $n_i(t)$ is the density of cell i at time t. The second term, $Q\Delta t$, is the product of the capacity and the time step size, which is the largest value of the inter-cell flow. The third term $\frac{w}{v_f}(N_{i+1} - n_i)$ includes the backward shockwave speed w, the free flow speed v_f, and the maximum density N in cell $i+1$. The maximum density N is calculated using the jam density K and the cell length ΔL, which is $N = K \times \Delta L$. The third term represents the available space in the next cell.

3.2 The Structure of the Kalman Filter

In the Kalman filter with CTM, BSM data including vehicle locations and speeds can be processed to obtain relevant traffic state information such as density, flow, and speed. The basic equations of the Kalman filter are shown in this section.

The prediction step in equation (5) is made up of the state vector $\mathbf{N}(t)$ and the control vector $\mathbf{Y}(t)$. The state vector includes the densities of all road segments. The reason why only the density is included in the state vector is that other traffic state variables, such as the flow and the speed, can be calculated using the density based on the fundamental diagram. The calculation of densities requires the knowledge of incoming flows and outgoing flows, which is included in control vector $\mathbf{Y}(t)$ in equation (4). The first element $y_0(t)$ is the input flow to the road, which is assumed to be the measured flow for equipped vehicle $y_0^p(t)$ divided by the penetration rate r.

$$\mathbf{N}(t) = \begin{bmatrix} n_1(t) \\ n_2(t) \\ \vdots \\ n_i(t) \end{bmatrix} \tag{3}$$

$$\mathbf{Y}(t) = \begin{bmatrix} y_0(t) \\ y_1(t) \\ \vdots \\ y_i(t) \end{bmatrix} \tag{4}$$

The relation between the state vector $\mathbf{N}(t)$ and the control vector $\mathbf{Y}(t)$ can be written as the form

in equation (5).

$$\mathbf{N}(t+1) = \mathbf{A_r}\mathbf{N}(t) + \mathbf{B_r}\mathbf{Y}(t+1) + \mathbf{Q_r}^* \tag{5}$$

In equation (5), the state vector at time $t+1$ is updated with the state vector at time t and the control vector at time $t+1$. $\mathbf{A_r}$ is an identity matrix. $\mathbf{B_r}$ is a matrix with i rows and $i+1$ columns with all elements equal to -1, 0 or 1. Its form is shown in equation (6). The matrices $\mathbf{A_r}$ and $\mathbf{B_r}$ are derived from the relation between the state vector and the control vector shown in equation (1). Matrix $\mathbf{Q_r}^*$ is the error term for the prediction.

$$\mathbf{B_r} = \begin{bmatrix} 1 & -1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & -1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & -1 \end{bmatrix} \tag{6}$$

Equation (7) shows the measurement step of the Kalman Filter. $\hat{\mathbf{N}}(t)$ is the measurement of densities. $\mathbf{C_r}$ is an identity matrix. $\mathbf{R_r}$ is the error term related with measurement, such as measurement errors from detectors. In this study, the relation between the measured density for all vehicles $\hat{\mathbf{N}}$ and the measured densities for equipped vehicles $\mathbf{N}_p$ can be represented by function $\mathbf{G}$.

$$\hat{\mathbf{N}}(t) = \mathbf{C_r}\mathbf{G}\left(\hat{\mathbf{N}}_p(t), \hat{\mathbf{v}}_p(t)\right) + \mathbf{R_r} \tag{7}$$

$$\mathbf{G}\left(\hat{\mathbf{N}}_p(t), \hat{\mathbf{v}}_p(t)\right) = \begin{cases} \frac{\hat{\mathbf{N}}_p(t)}{r}, & \hat{\mathbf{v}}_p(t) \geq 0.9 v_\mathrm{f} \\ \frac{wK\Delta l}{\hat{\mathbf{v}}_p(t)+w}, & \hat{\mathbf{v}}_p(t) < 0.9 v_\mathrm{f} \end{cases} \tag{8}$$

In equation (8), when the measured average speed of probe vehicles $\hat{\mathbf{v}}_p(t)$ is greater or equal to 0.9 times the free-flow speed, the traffic is considered uncongested. Then the measured density is the measured number of equipped vehicles $\hat{\mathbf{N}}_p(t)$ divided by the assumed penetration rate r. Otherwise, the road is congested. The density is calculated by finding the corresponding value on the fundamental diagram using the measured speed of equipped vehicles. If no speed data is available at the current time step, the measurement at the previous time step is used. To make the fundamental diagram accurately represent the dynamics of traffic flow on the road, the values of capacity Q, the free-flow speed v_f, and the jam density K should be calibrated in advance or updated in real-time. The parameter calibration will be discussed in section 4. The covariance matrix $\mathbf{P_r}$ can be updated using equation (9). As matrix $\mathbf{A_r}$ is a identity matrix, matrix $\mathbf{P_r}$ can be

calculated by adding up matrix $\mathbf{A_r}$ and the covariance matrix $\mathbf{Q_r}^*$.

$$\mathbf{P_r}(t+1) = \mathbf{A_r}\mathbf{P_r}(t)\mathbf{A_r}^\intercal + \mathbf{Q_r}^* \tag{9}$$

The Kalman Gain $\mathbf{K_r}$, which serves as the weight of measurement over prediction in the estimation, is calculated by equation (10).

$$\mathbf{K_r}(t+1) = \mathbf{P_r}(t+1|t)\mathbf{C_r}^\intercal \left[\mathbf{C_r}\mathbf{P_r}(t+1|t)\mathbf{C_r}^\intercal + \mathbf{R_r}\right]^{-1} \tag{10}$$

Using Kalman Gain $\mathbf{K_r}$, the values of predictions and measurements can be integrated and the covariance matrix $\mathbf{P_r}$ can be updated, as shown in equations (11) and (12).

$$\mathbf{N}(t+1|t+1) = \mathbf{N}(t+1|t) + \mathbf{K_r}(t+1)\left[\hat{\mathbf{N}}(t+1) - \mathbf{C_r}\mathbf{N}(t+1|t)\right] \tag{11}$$

$$\mathbf{P_r}(t+1|t+1) = \mathbf{P_r}(t+1|t) - \mathbf{K_r}(t+1)\mathbf{C_r}\mathbf{P_r}(t+1|t) \tag{12}$$

With the segment length ΔL, the density can be converted to density $k_i(t) = n_i(t)/\Delta L$. With the value of k, the flow rate and speed can also be estimated.

3.3 Parameter Calibration

CTM relies on accurate fundamental diagram parameters, which vary depending on the road and the weather conditions. It is necessary to estimate parameters to make them reflect the traffic flow dynamics on the target road. These parameters include the capacity Q, the free-flow seed v_f, the backward shockwave speed w, and the jam density K. To simplify the process of parameter calibration, the value for the jam density K is first calculated by the average car length ℓ, as shown in (13). Then the parameter calibration will not update the value for the jam density.

$$K = \frac{1}{\ell} \tag{13}$$

It is assumed that all road segments on a road with the same number of lanes share the same parameters in the fundamental diagram. The parameter calibration uses a lower frequency to update the values of parameters than the traffic state estimation process mentioned in section 3.

3.3.1 Measurement of Parameters

The values of the free-flow speed, the capacity, and the shockwave speed should be measured from the BSM data.

If the road is congested during the previous time step, and the average speed at the current time step is much larger than that in the previous time step, then the current average speed, $\bar{v}(t)$, will be used as the new estimation of the free-flow speed. For BSM data, it is hard to directly get measurements of the capacity and the shockwave speed because BSM data only includes the density and flow related to equipped vehicles. However, it is assumed that the average speed of equipped vehicles can represent the speed for the entire road segment in the road section. If there is no huge drop in the measured capacity, the free-flow speed can be extracted by getting the larger value between the average speed at the current time and the value of free-flow speed at the previous time step, as shown in equation (14). If a drop in the measured capacity is observed, the measured value for free-flow speed can be decreased, as shown in equation (15).

$$\hat{v}_{\mathsf{f}}(t) = \mathsf{max}\left(\hat{v}_{\mathsf{f}}(t-1), \bar{v}(t)\right) \tag{14}$$

$$\hat{v}_{\mathsf{f}}(t) = \bar{v}(t) \tag{15}$$

In the triangular fundamental diagram, the traffic has congested and uncongested conditions. When the average speed $\bar{v}(t)$ is greatly less than the free-flow speed v_{f}, the traffic is congested. Otherwise, the road is uncongested. When the road is congested, it is possible to estimate the backward shockwave speed $\hat{w}$ using equation (16). ℓ is the average car length, and τ is the average reaction time of drivers. The time when a shockwave can be observed is not common so it is difficult to get a good estimation of the average reaction time of drivers. If τ is not available, the headway between two consecutive vehicles is used. The measured value for capacity $\hat{Q}$ can be calculated based on the value of the shockwave speed, as shown in (17).

$$\hat{w} = \frac{\ell}{\tau} \tag{16}$$

$$\hat{Q} = \left(\frac{1}{v_{\mathsf{f}}} + \frac{1}{\hat{w}}\right) K \tag{17}$$

When the road is uncongested, the capacity is estimated first. Its value can be estimated with equation (18). In equation (18), $\min(\bar{h})$ represents the smallest headway in the section. The capacity is the largest number of vehicles that can pass a road section in an hour, which is equal to the number of smallest headways in an hour. A smaller headway corresponds to a larger flow rate and capacity. Then the backward shockwave speed $\hat{w}$ can be calculated based on the value of the capacity, as shown in (19). When the market penetration of equipped vehicles is low, the case in which one equipped vehicle follow another equipped vehicle should be found to extract

the headway. If there is no such case in the current time step, the value of the capacity will not be updated.

$$\hat{Q} = \frac{3600}{\min(\bar{h})} \tag{18}$$

$$\hat{w} = \frac{K v_f}{\hat{Q} v_f - K} \tag{19}$$

3.3.2 The Structure of Kalman Filter in Parameter Calibration

In this study, the update of parameters also uses a Kalman filter structure. $\mathbf{F}(t)$ is the vector of parameters at time t.

$$\mathbf{F}(t) = \begin{bmatrix} Q(t) \\ v_f(t) \\ w(t) \end{bmatrix} \tag{20}$$

Equation (21) shows the updating process of these three parameters. $\mathbf{A_c}$ is an identity matrix because we assume that the prediction of parameters at time $t+1$ is just the values of parameters at time t, and the parameter value is a random walk.

$$\mathbf{F}(t+1) = \mathbf{A_c}\mathbf{F}(t) + \mathbf{Q_c}^* \tag{21}$$

The measurement vector $\hat{\mathbf{F}}$ is consisted of the measurements of the capacity, the free-flow speed and the backward shockwave speed. The method of getting the measurement is mentioned in section 3.3.1.

$$\hat{\mathbf{F}}(t) = \begin{bmatrix} \hat{Q}(t) \\ \hat{v}_f(t) \\ \hat{w}(t) \end{bmatrix} \tag{22}$$

Kalman Gain is used to integrate these two parts, which is calculated by equation (23). In this equation, $\mathbf{C_r}$ is a 3×3 identity matrix, and the measurement of one parameter is assumed to be independent of those of other parameters.

$$\mathbf{K_c}(t+1) = \mathbf{P}_c(t+1|t)\mathbf{C_r}^\intercal \left[\mathbf{C_r}\mathbf{P}_c(t+1|t)\mathbf{C_r}^\intercal + \mathbf{R_c}\right]^{-1} \tag{23}$$

Using Kalman Gain $\mathbf{K_c}$, the values of predictions and measurements are integrated and the covariance matrix is updated with equations (24) and (25).

$$\mathbf{F}(t+1|t+1) = \mathbf{F}(t+1|t) + \mathbf{K_c}(t+1)\left[\hat{\mathbf{F}}(t+1) - \mathbf{C_c}\mathbf{F}(t+1|t)\right] \tag{24}$$

$$\mathbf{P_c}(t+1|t+1) = \mathbf{P_c}(t+1|t) - \mathbf{K_c}(t+1)\mathbf{C_c}\mathbf{P_c}(t+1|t) \quad (25)$$

4 Experiments with a Simulator Built with CTM

CTM simulators were used to generate multiple databases to test the proposed algorithm. In the test using CTM simulators, the research team tended to test the basic structure of the code, so a relatively high market penetration of 60% was used. In the test using the microscopic simulator, market penetrations of 20% and 40% were used.

4.1 Test with CTM simulator

Two test databases are generated by simulators based on CTM. The output of this simulator includes vehicle IDs, vehicle locations, and vehicle speeds at each time step. Vehicle locations should be converted to the ID of the road segment that contains the vehicle. The output file only has the trajectories of a subset of the total vehicles. For example, if the market penetration is 60%, then the output file only includes the trajectories of 60% of the vehicles.

4.1.1 Test 1 with known parameters

Figure 2 shows a test with all parameters known. The orange line is the actual density, and the blue line is the estimated density. The penetration rate is 60%. The measured density varies a lot and reaches zero at some time steps. It is because the input flow rate is 2 vehicles per time step and BSMs only provide data from 60% of the vehicles. Figure 2 reflects that if parameter values and the time of the incident are known, the CTM can be a good representation of the dynamics of the traffic flow generated by this simulator. For example, when there is a capacity drop at $t = 40$, the density estimation is close to the actual density of 8 vehicles, even though the measured density of probe vehicles is 4. The prediction step based on CTM corrects the measurement of 4 vehicles (shown in Figure 3), and the final estimation for this road segment at time step 46 is calculated to be 7.6 (round to 8). The accuracy of the Kalman filter is higher when there is an incident compared with the free-flow condition. When there is an incident, the actual density will be much higher than the measured density of equipped vehicles and the Kalman filter will put more weight on the prediction step.

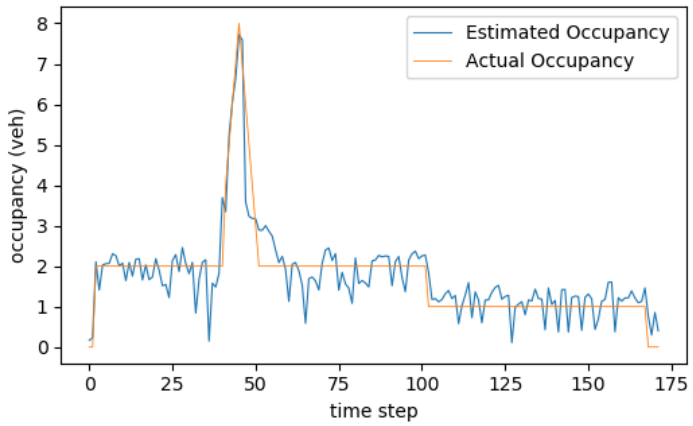


Figure. 2: Test 1 with known parameters (estimated density vs actual density)

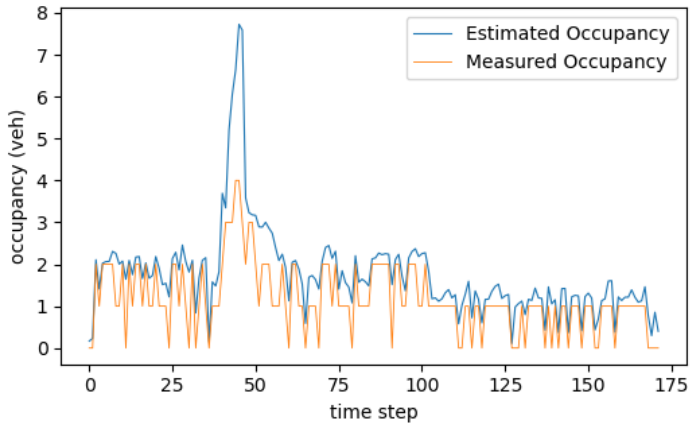


Figure. 3: Test 1 with known parameters (estimated density vs measured density)

Figure 5 shows the estimated speed and measured speed of equipped vehicles. Figure 4 shows the estimated speed and the actual speed. The measured speed is slightly different from the actual speed. When there is no equipped vehicle at the current time step, the measured speed at last time step is used as the measured speed at the current time step, so it is possible that the actual speed has a big change but the measured speed remains the same. When there are few equipped vehicles in a road segment that behave differently from all the other vehicles in the same road segment, the measured speed and the actual speed will be different. In Figure 4, when the average speed is in free-flow speed, the Kalman filter sometimes overestimates the average speed. When there is a drop in the travel speed, the estimated speed is close to the actual speed. Compared with the density, the speed estimation has higher density. In practice, the travel speed is used to determine the level of service of the road. Traffic engineers are more focused on the

speed estimation of the congestion condition than the free-flow condition. Therefore, the speed estimation in the Kalman filter can provide useful information in practice.

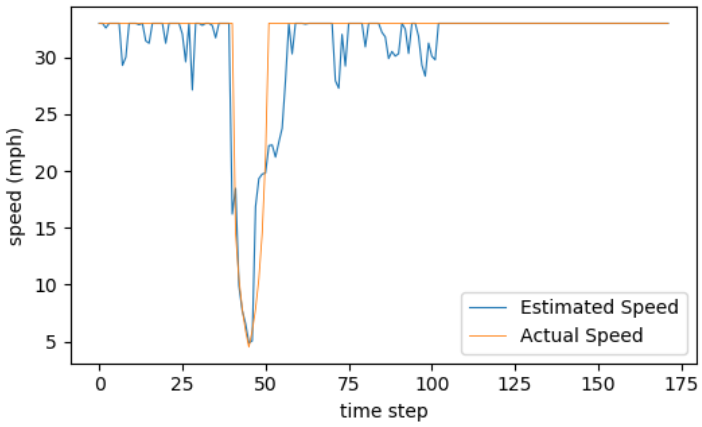


Figure. 4: Test 1 with known parameters (estimated speed vs actual speed)

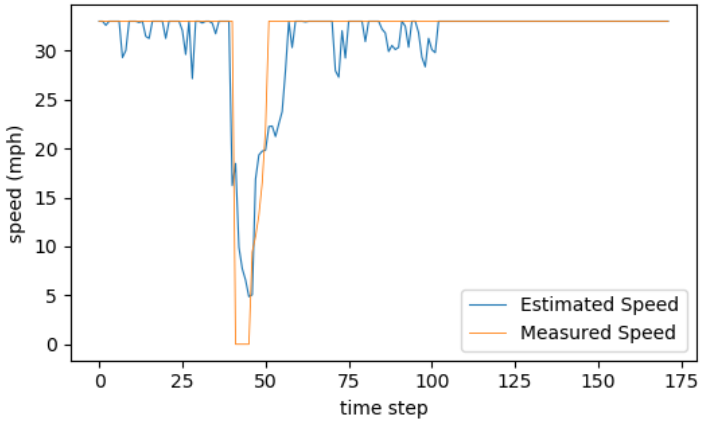


Figure. 5: Test 1 with known parameters (estimated speed vs measured speed)

4.1.2 Test 2 with unknown parameters

In test 2, another CTM-based simulator was used to generate the test database. The values of parameters are unknown and the algorithm for parameter calibration runs simultaneously with the Kalman filter to get traffic state estimations. The market penetration rate is 65%. The results show that the free-flow speed is easier to calibrate than the capacity or congested wave speed. In Figure 8, the initial value of the free-flow speed is 30 miles per hour, but it converges to the actual value of free-flow speed in a short time. In this test, the capacity and the shockwave speed do not converge to their right values. The trajectory data generated by the CTM-based simulator does

not include enough information about the time gap or spacing between two vehicles. For capacity, it is not reasonable to use the maximum value of the flow rate for the value of capacity because the measured flow rate is only for equipped vehicles. For the backward shockwave speed, the calculation needs the reaction time between vehicles, which requires the case where two vehicles are close to each other. If there is no such case observed, the shockwave speed cannot be fully calibrated. In practice, the backward shockwave speed can also be calibrated by searching cases where two adjacent probe vehicles both have a drop in travel speed. With the time gap between two speed-drops and the distance between these two probe vehicles, the backward shockwave speed can be estimated.

If the parameters for CTM are not accurately calibrated, the prediction step of the Kalman filter will have limited accuracy. In Figure 6, the line representing estimation always follows the line representing the measurement. The estimation highly relies on the measurement step of the Kalman filter, which cannot represent the density including all vehicles on the road. In Figure 7, from time step 75 to time step 183, the actual density is always larger than the estimated density.

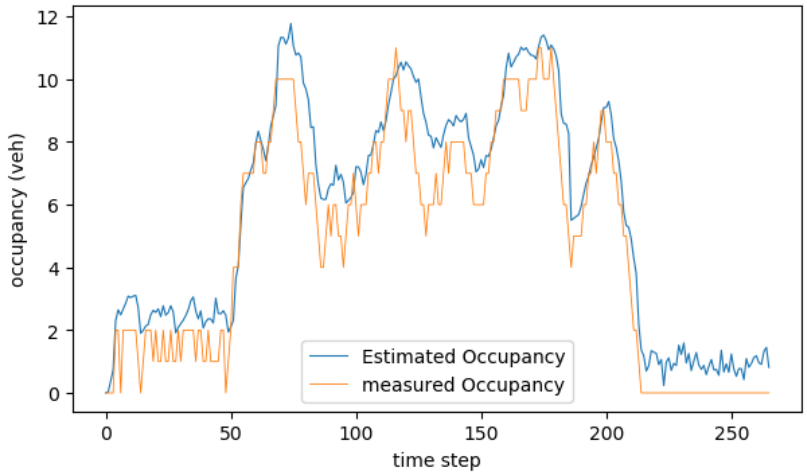


Figure. 6: Test 2 with known parameters (estimated density vs measured density)

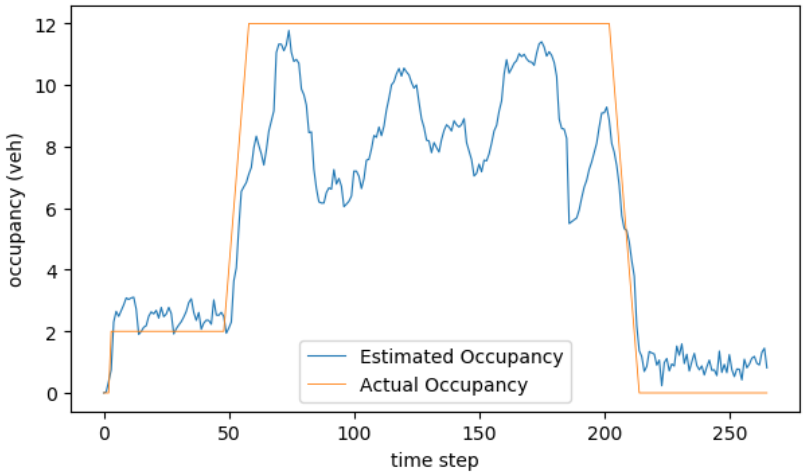


Figure. 7: Test 2 with known parameters (estimated density vs actual density)

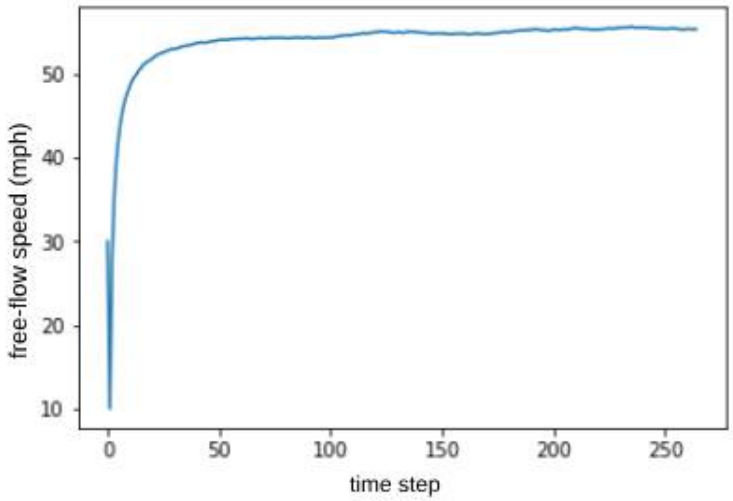


Figure. 8: Calibration of free-flow speed in Test 2

5 Simulation-generated BSM

Real sensor information is never perfect, often containing: missing or imputed data, random and periodic noise, and measurement artifacts generated either by the sensor or as a result of the method by which the system is handling the sensor output. We aims to create analytical methodologies that translate high resolution abstract sensor information into aggregate metrics quantifying real traffic conditions. To develop them, the methodologies needed to first be tested against an "unrealistically" true ground truth, with no unknowns, which is only possible through simulation. The simulation data, in addition to being unpolluted by missing data or noise, was provided so that we could continue on methodologies while the real world data from the CV Testbed was being gathered and reduced. By using pure data gathered from simulation, methodologies could be tested for accuracy without the issues that a real world data set might have, to ensure that they were sound before moving to real world data.

5.1 Undisputed ground truth

To provide a zero-unknown ground truth, the Minnesota Traffic Observatory selected a very detailed existing simulation model, which was created in Aimsun. This simulation was of the City of Richfield and surrounding areas, comprising a complex network of all types of roadways (arterials, highways, ramps, etc.) that had been accurately calibrated on its creation.

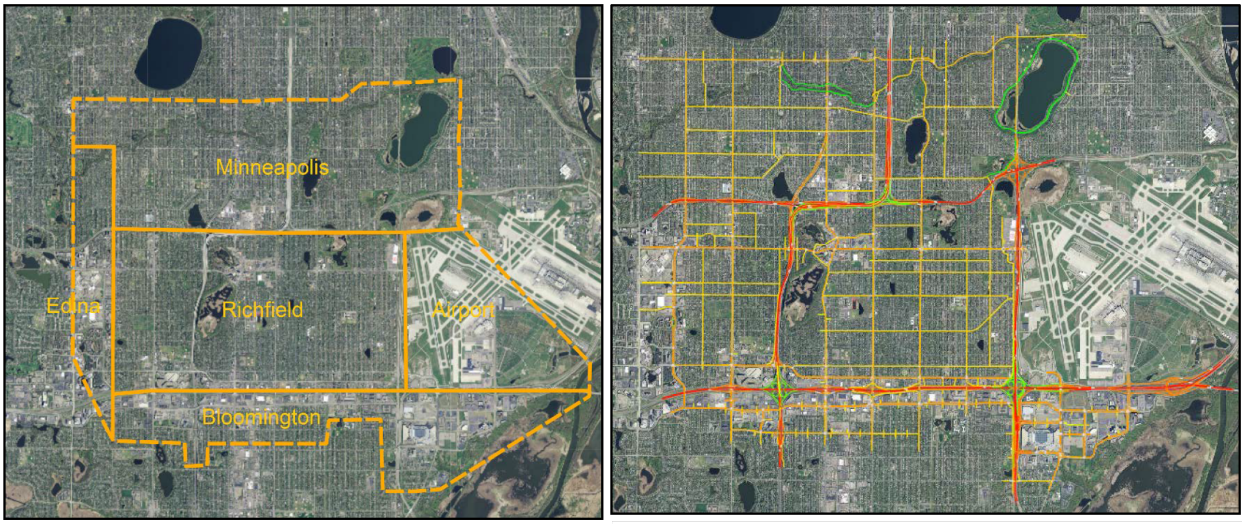


Figure. 9: Richfield simulation area (left) and road type (right)

To use this simulation to emulate BSMs, a simulator plug-in was developed that could work on any stretch of simulated roadway. This plug-in allows for the introduction of Connected Vehicles

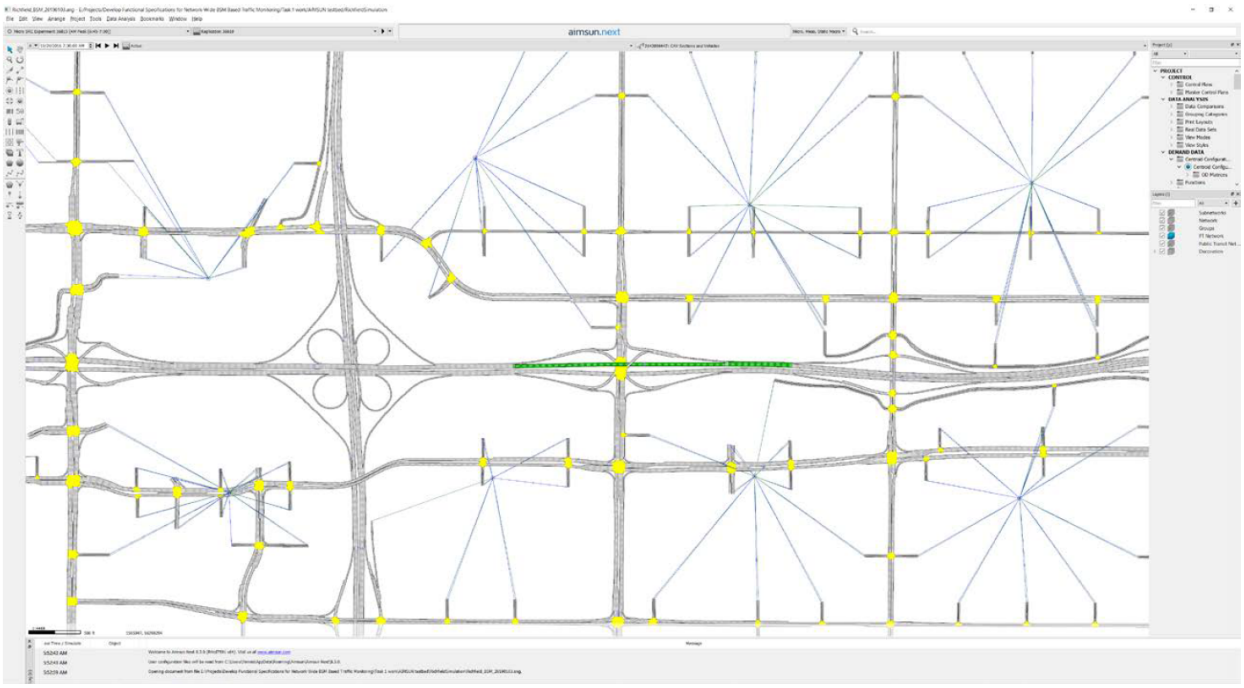


Figure. 10: A simulated section of roadway along I-494 in Richfield containing emulated RSUs (green highlight)

(CVs) and/or Connected Automated Vehicles (CAVs) to the vehicle population; it also emulates the deployment of Road Side Units (RSUs) on the desired road sections. The plug-in tracks selected simulated vehicles and extracts BSM information as long as they are traversing selected emulated RSU roadway segments. That way, the plug-in can emulate any length of instrumented roadway as well as segmented pieces with intermediate non-instrumented parts. BSM data will only be available for equipped vehicles when they are in instrumented sections.

Figure 10 shows a section of I-494 and Lyndale Ave in the Richfield simulation. The green stretch of roadway in the middle has the plug-in to emulate RSUs active.

Though real-world BSM data will only contain vehicle ID, speed, latitude/longitude, and vehicle heading, the plug-in captured more information than that as determined by Aimsun parameters. The simulation creates two records for each vehicle, for static and dynamic states. The static vehicle record, shown in Table 2.1, contains information about the vehicle that was unaffected by instrumentation. The dynamic vehicle record, shown in Table 2.2 contains information for each simulation step when the vehicle was located in an instrumented section.

Table 2: Static vehicle information record

Column Name	Description
vehicle_id	The vehicle identifier.
vehicle_type	Vehicle type name (string).
vehicle_length	Vehicle length (m or feet).
vehicle_width	Vehicle width (m or feet).
max_desired_speed	Maximum desired speed of the vehicle (km/h or mph).
max_acceleration	Maximum acceleration of the vehicle (m/s2 or ft/ s2).
normal_deceleration	Maximum deceleration of the vehicle that can apply under normal conditions (m/s2 or ft/s2).
max_deceleration	Maximum deceleration of the vehicle that can apply under special conditions (m/s2 or ft/ s2).
speed_acceptance	Degree of acceptance of the speed limits.
min_following_distance	Distance that the vehicle keeps between itself and the preceding vehicle (meters or feet).
give_way_time	Time after which the vehicle becomes more aggressive in give-way situations (seconds).
guidance_acceptance	Level of compliance of the vehicle to guidance indications.
enrouted	0 means vehicle will not change path enroute, 1 means vehicle will change path enroute depending on the percentege of enrouted vehicles defined.
keep_fast_lane	Means vehicle keep fast lane during overtaking.
min_headway	Minimum headway to keep with its leader.
sensitivity_factor	Estimation of the acceleration of the leader.
reaction_time	Reaction time of the vehicle (seconds).
reaction_time_at_stop	Reaction time at stop of the vehicle (seconds).
reaction_time_at_traffic_light	\Rreaction time of the vehicle when stopped the first one of the queue in a traffic light (seconds).
origin_centroid	Identifier of centroid origin of the vehicle, when the traffic conditions are defined by an O/D matrix.
destination_centroid	Identifier of centroid destination of the vehicle, when the traffic conditions are defined by an O/D matrix.
exit_section_id	Identifier of exit section destination of the vehicle, when the destination centroid uses percentages as destination (otherwise is –1) and the traffic conditions are defined by an O/D matrix.
system_generation_time	The absolute generation time of the vehicle into the system. If no virtual queue found in its entrance section it will be the same as the SystemEntranceT.
system_entrance_time	The absolute entrance time of the vehicle into the system, that is into its entrance section. If no virtual queue found in its entrance section it will be the same as the SystemGenerationT.

Table 3: Dynamic vehicle information record

Column Name	Description
vehicle_id	The vehicle identifier.
simulation_time	The simulation time of this measurement.
section_id	The identifier of the section where the vehicle is located.
segment_id	The segment number of the section where the vehicle is located (from 0 to n-1).
lane_number	The lane number in the segment (from 1, the rightmost lanet, to N, the leftmost lane).
current_pos_section	The position inside the section (distance in m or feet from the beginning of the section).
distance_end_section	The distance to the end of the section (m or feet).
world_pos_x	X world coordinate of the middle point of the front bumper of the vehicle.
world_pos_y	Y world coordinate of the middle point of the front bumper of the vehicle.
world_pos_z	Z world coordinate of the middle point of the front bumper of the vehicle.
world_pos_x_rear	X world coordinate of the middle point of the rear bumper of the vehicle.
world_pos_y_rear	Y world coordinate of the middle point of the rear bumper of the vehicle.
world_pos_z_rear	Z world coordinate of the middle point of the rear bumper of the vehicle.
current_speed	The current speed of the vehicle (in km/h or mph).
distance_traveled	The total distance travelled (m or feet) by the vehicle.
section_entrance_time	The absolute entrance time of the vehicle into the current section.
current_stop_time	The current stop time.

6 Experiments with simulated-based BSMs

In this test, the microscopic simulation software AIMSUN was used to generate the database. Compared with the CTM simulator, the microscopic simulation model can generate drivers with more realistic driving behaviors and provide data that is more similar to actual BSM data. The preparation of simulation-based BSMs is introduced in section 5.

6.1 Ground truth database preparation

To apply the database generated by the microscopic simulation model for the traffic state estimation, some data preprocessing works should be finished first.

If the location of the origin is known, the travel distance of a vehicle cannot be directly calculated using the current location (x,y) of this vehicle because the road section may be curved. It is more accurate to get the cumulative traveled distance along its trajectory, as shown in equation (26).

$$\text{Travel Distance} = \sum_{k=1}^{t-1} \sqrt{(x(k+1)-x(k))^2 + (y(k+1)-y(k))^2} \tag{26}$$

The ground truth database needs to be prepared in advance to compare with the estimated results. We use the traffic state directly extracted using BSMs from 100% equipped vehicles as the actual traffic state. We did not use the measurements directly from the simulation because we need to get the estimated traffic states for each cell on a road section, for example, the number of vehicles, the average space-mean speed, and exiting flow of each cell. Existing tools in AIMSUN cannot provide us measurements for each cell so we need to extract traffic states using the 100% BSMs using methods from existing studies. The trajectories of vehicles in the output of AIMSUN have a resolution of 0.1 second, but the required traffic state measurement is for a time interval that is equal to the time step of CTM. The method that converts 100% BSMs to traffic state measurements are from the study of (11), as shown in equations below. Equation (27) defines the area of a time-space zone, which is the product of the cell length ΔL and the time step size Δt. Equations (28) and (29) define the average flow and average density for a time-space zone respectively, which requires the travel distances x_i and the travel times t_i of all vehicles in a time-space zone. Equation (30) defines the average speed as the ratio of the average flow and the average density. Figure 11 shows the time-space zone used in Eddie's definition.

$$|A| = \Delta t \times \Delta L \tag{27}$$

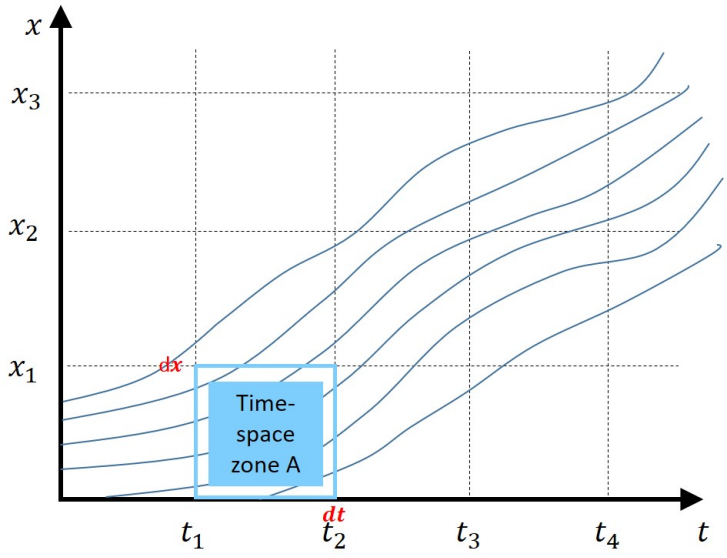


Figure. 11: Time-space zone

$$q = \frac{\sum_i x_i}{|A|} \tag{28}$$

$$k = \frac{\sum_i t_i}{|A|} \tag{29}$$

$$v = \frac{\sum_i x_i}{\sum_i t_i} \tag{30}$$

With the ground truth data and the estimated traffic states, estimation error can be calculated by equation (31), where the numerator is the absolute value of the difference between the estimated value $\hat{x}$ and the actual value $\tilde{x}$, and the denominator is the actual value $\tilde{x}$.

$$\text{Error} = \frac{|\hat{x} - \tilde{x}|}{\tilde{x}} \tag{31}$$

6.2 Test databases with simulated-based BSMs

The microscopic simulation model was calibrated with freeway detector data and traffic demand data. In Figure 12, two roads in the red box are the target roads whose vehicle trajectories are extracted during the simulation. These two roads are northbound and southbound sections of freeway I–35. API is used to extract the BSMs during a 45-min simulation corresponding to the morning pick hour (6:45 am – 7:30 am). In the simulation, the northbound section of I–35 is not congested while the southbound section of I–35 is congested. Two databases are built separately for these two sections of I–35. After the database including 100% of BSMs is generated (used to

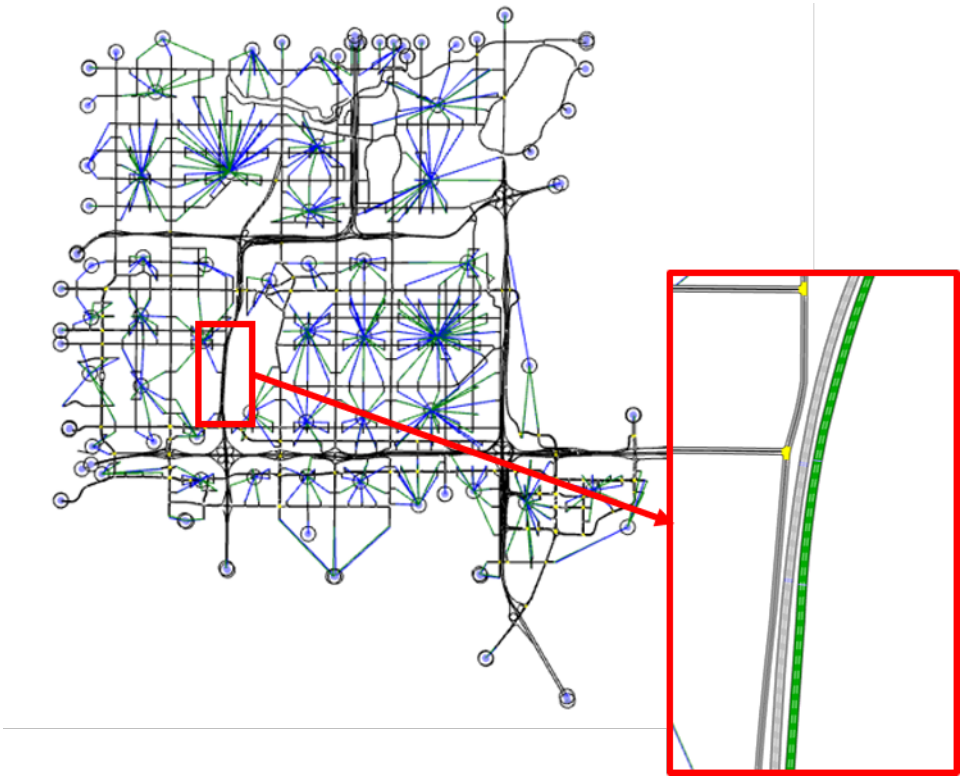


Figure. 12: Target road sections in AIMSUN

compare the estimation to the actual traffic state), the other four datasets are generated including 20%, 40%, 60%, and 80% of BSMs. To generate these datasets with partial trajectories, we generate a random number between 0 and 1 for each vehicle. If the random number is smaller than the set penetration rate, the BSMs of this vehicle is stored, otherwise, the vehicle BSMs are ignored. There are 8 databases generated in total. The sizes of these 8 databases, as well as the files for ground truth data, are shown in table 4.

Table 5: Estimated traffic flow parameters for experiment 1 and 2

Parameter	Value
Free-flow speed v_f	60.9 miles per hour
Capacity Q	2200 vehicles per hour per lane
Shockwave Speed w	17.2 miles per hour
Jam density K	234 vehicles per mile per lane

Table 4: File sizes of BSM databases from microsimulation models

Congested/uncongested	Penetration rate	File size	Row number
Uncongested	20%	74.5 MB	416586
Uncongested	40%	154 MB	863222
Uncongested	60%	231 MB	1291084
Uncongested	80%	333 MB	1862971
Uncongested	100%	384 MB	2148561
Congested	20%	107 MB	743698
Congested	40%	213 MB	1474930
Congested	60%	319 MB	2202720
Congested	80%	426 MB	2941030
Congested	100%	650 MB	3653996

The database used for parameter calibration is extracted from another 45-minute simulation on a road section on highway 494. Table 5 shows the estimated values of parameters.

6.3 Uncongested scenario

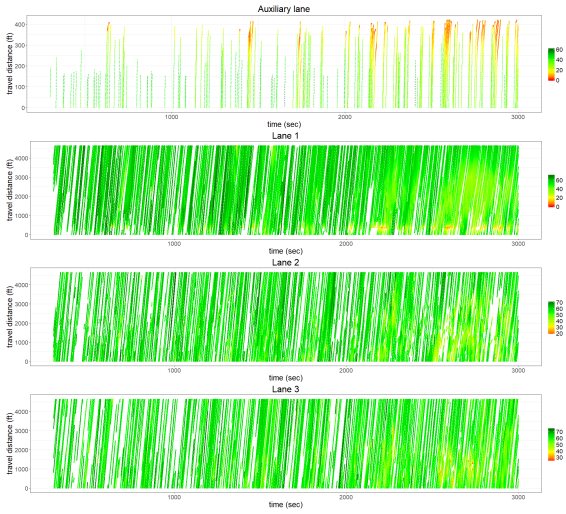
Figures 13 and 14 show the BSMs generated from the simulation under uncongested condition with different penetration rates. In Figures 13 and 14, most data points except the data points for the auxiliary lane are in green, which indicates most vehicles move with relatively high speed during the 45-min time interval. In the last 15 minutes, congestion can be observed but most vehicles travel with a speed higher than 40 mph.

Figures 15, 16, and 17 show the traffic state estimation results with different penetration rates compared with the actual traffic states. For densities, speeds, and flows, the ranges of the y-axis in figures are 0 to 35 (vehicles), 0 to 65 (mph), and 0 to 15 (veh) respectively.

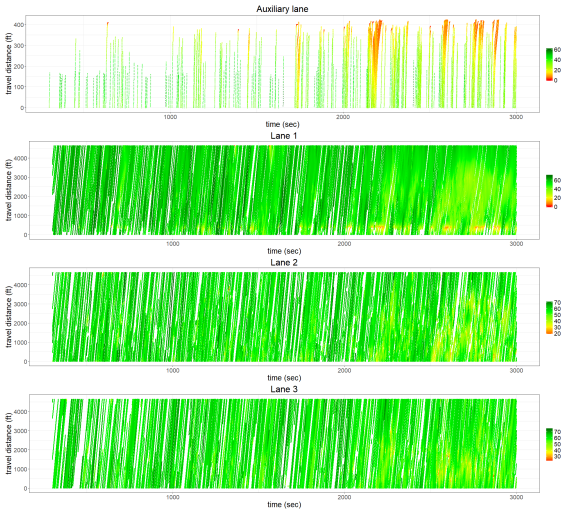
Before time step 120, all cells in this road section are not congested and have cell densities that are smaller than 15 vehicles. After time step 120, larger flows enter the road section through cell

1, so we can observe the cell color of the cell 1 becomes red. As large traffic flow passes through the road section, the cell color in the heatmap of the density from cell 2 to cell 8 becomes orange in order. After time step 300, as the road section becomes more congested, an orange area forms at the right side of the heatmap of the density. The Kalman filter can capture the formation and dissipation of the orange area in the heatmap of the density even with a low penetration rate. For the heatmap of the speed, as vehicle speeds can be directly measured from BSMs, the heatmap of estimated speed is similar to the heatmap of actual speeds. For the flow estimation, we can observe that the heatmap of estimated flow can capture the trend when input flow becomes larger after time step 120.

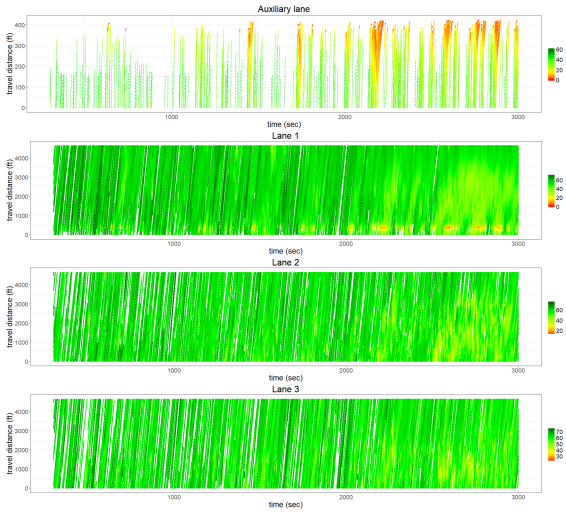
(a) 20% penetration



(b) 40% penetration



(c) 60% penetration



(d) 80% penetration

Figure. 13: BSMs from microscopic simulation model (uncongested condition)

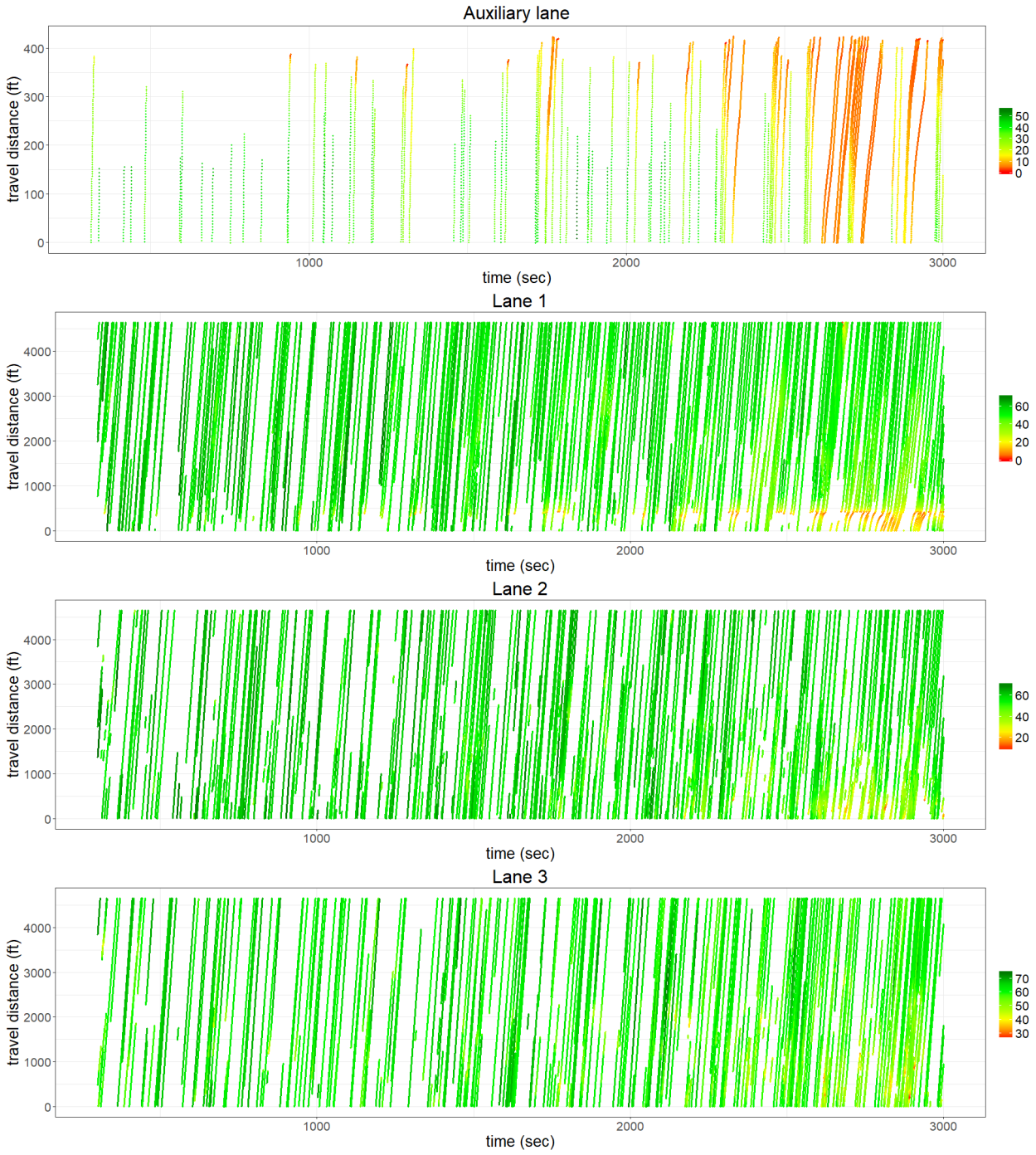
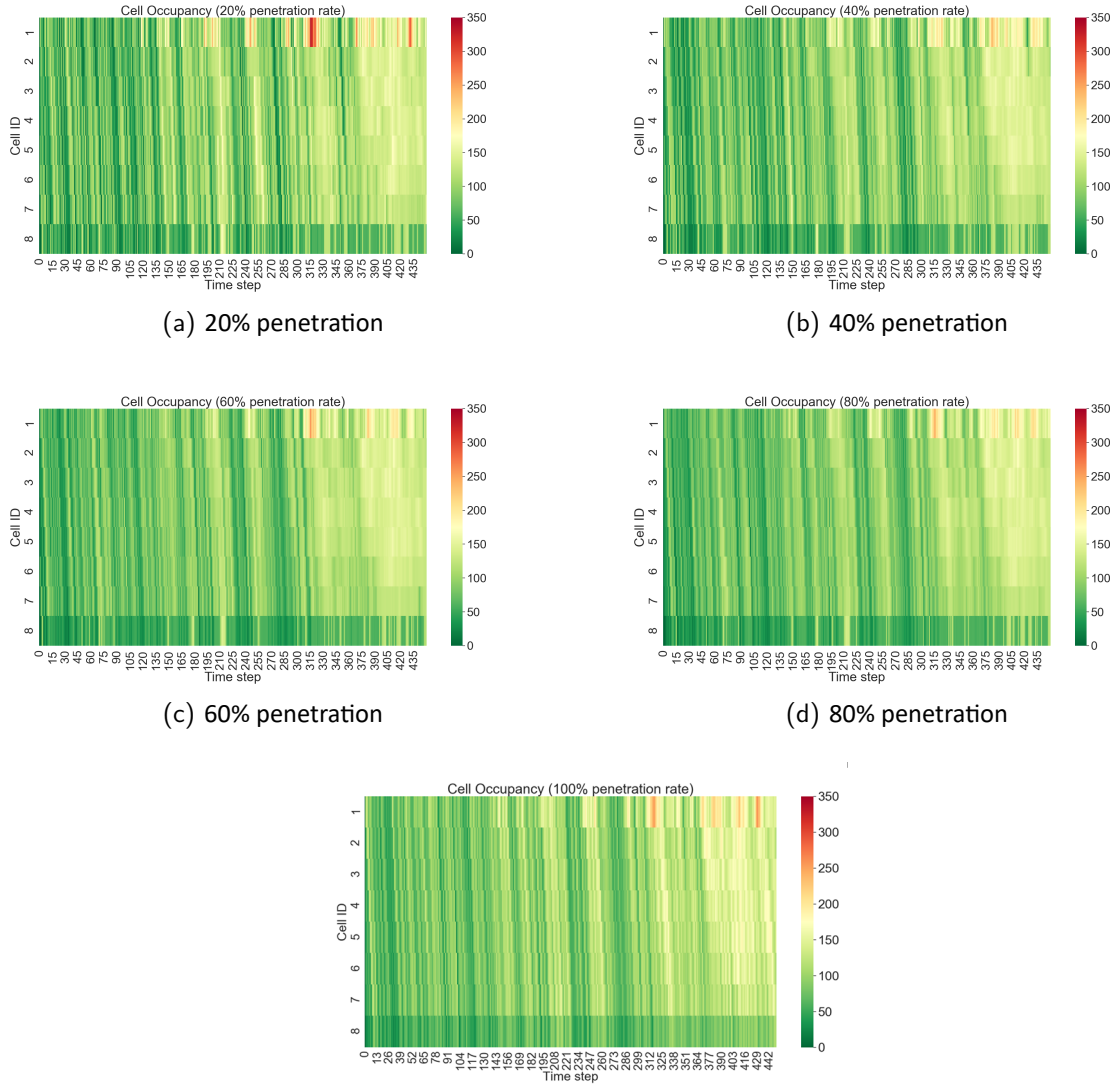


Figure. 14: 100% BSMs from microscopic simulation model (uncongested condition)

6.3.1 Traffic state estimation results



(a) 20% penetration

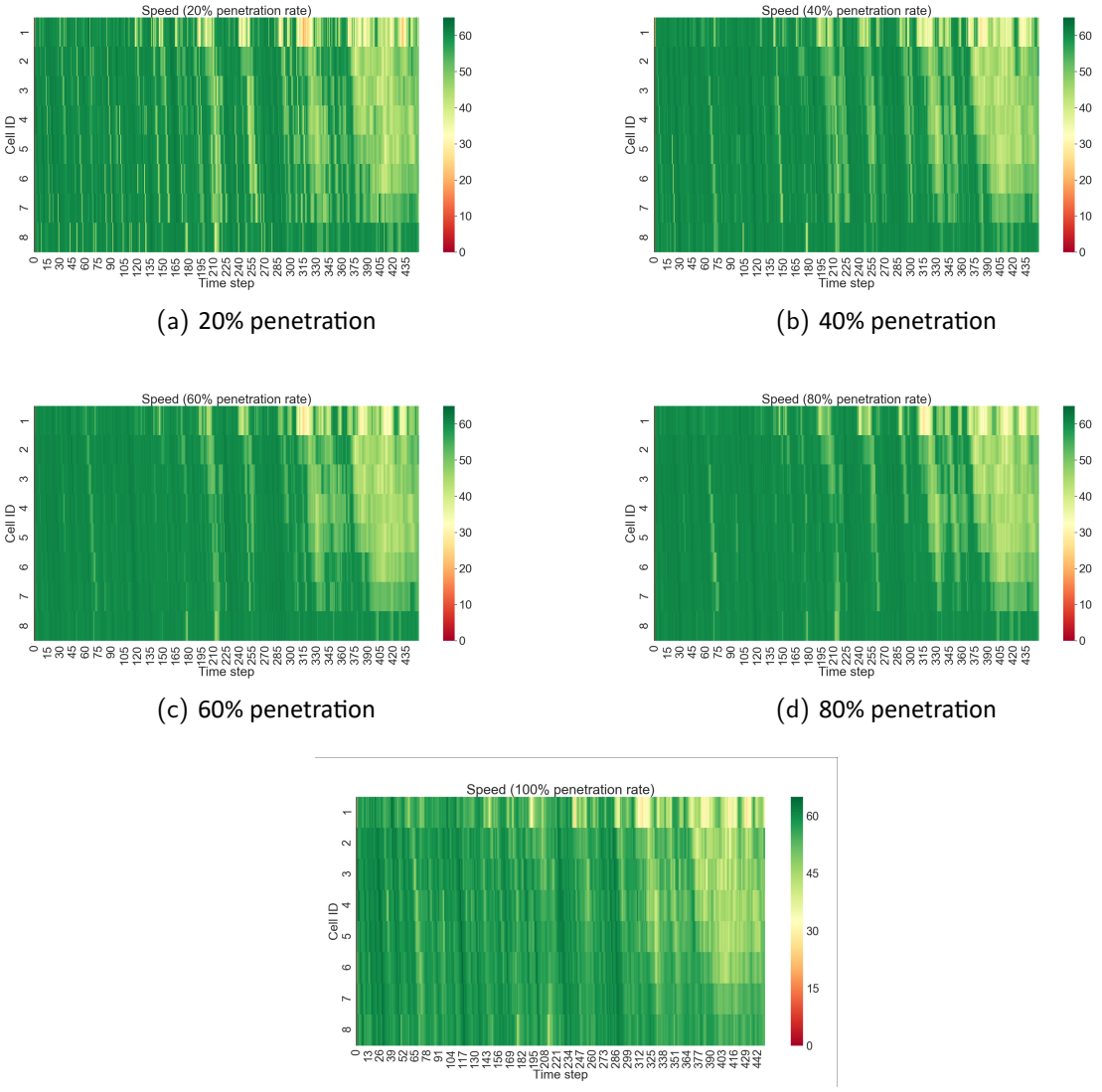
(b) 40% penetration

(c) 60% penetration

(d) 80% penetration

(e) 100% penetration

Figure. 15: Density estimation under different penetration rates with BSMs from microsimulation data (uncongested condition)



(a) 20% penetration

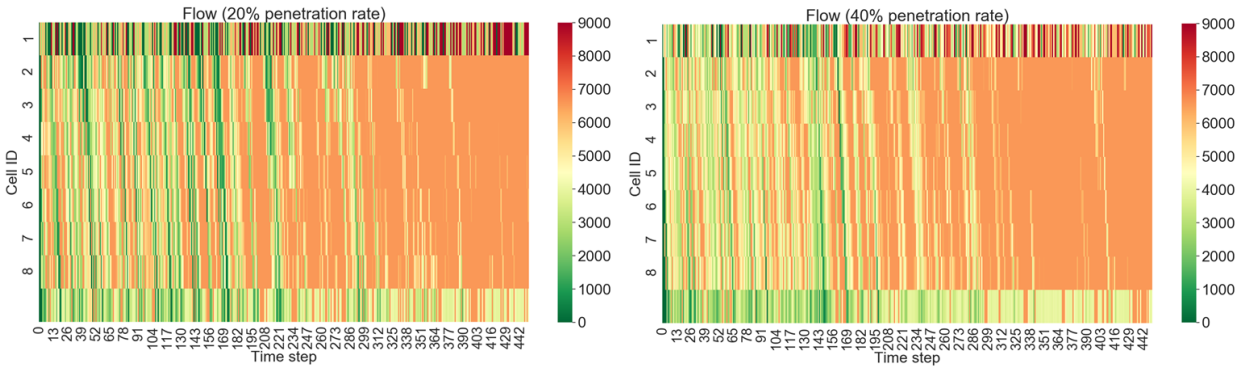
(b) 40% penetration

(c) 60% penetration

(d) 80% penetration

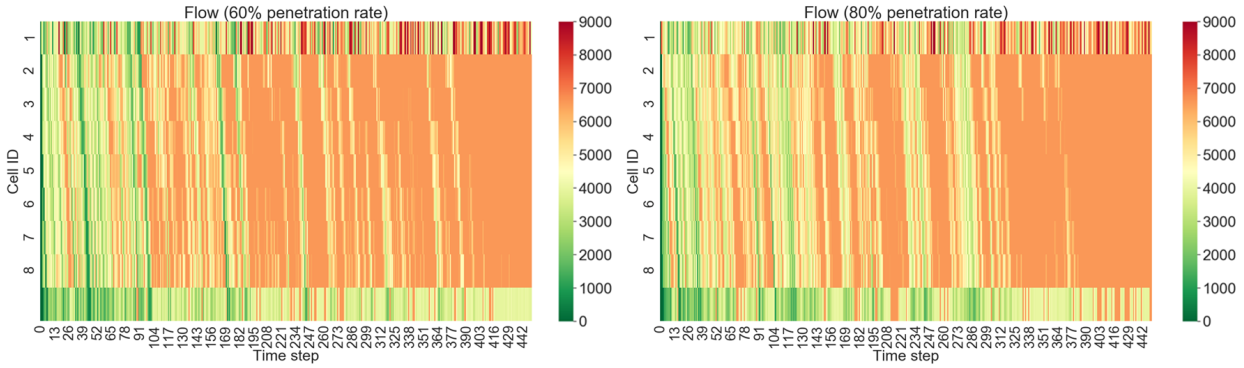
(e) 100% penetration

Figure. 16: Speed estimation under different penetration rates with BSMs from microsimulation data (uncongested condition)



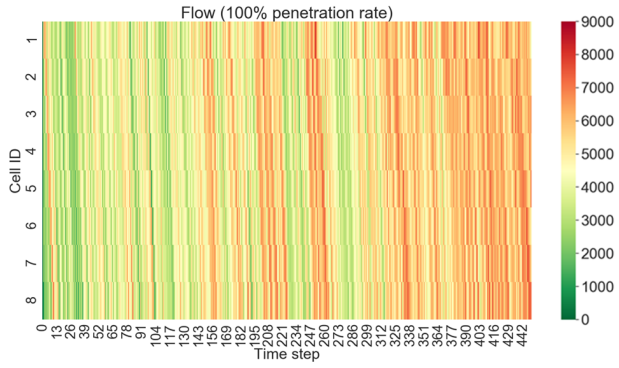
(a) 20% penetration

(b) 40% penetration



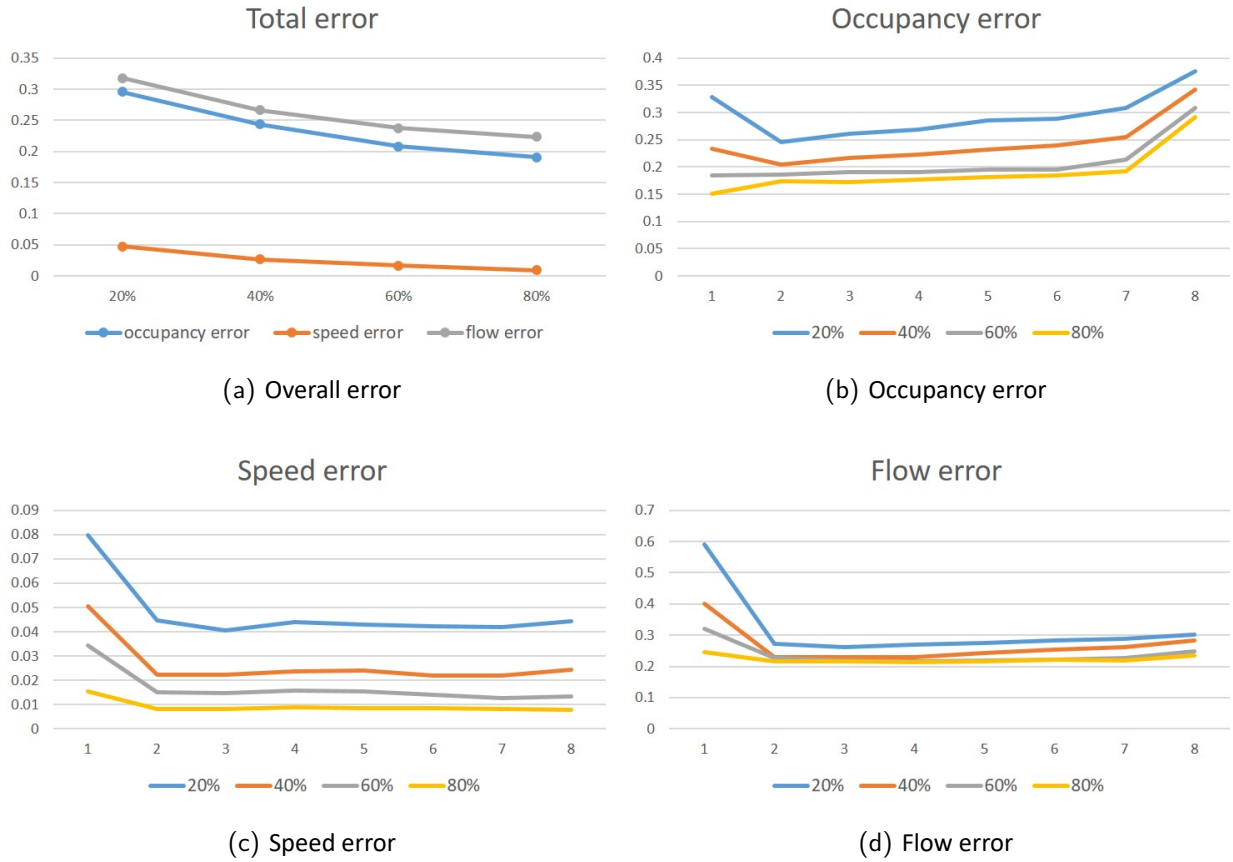
(c) 60% penetration

(d) 80% penetration



(e) 100% penetration

Figure. 17: Flow estimation under different penetration rates with BSMs from microsimulation data (uncongested condition)



(a) Overall error

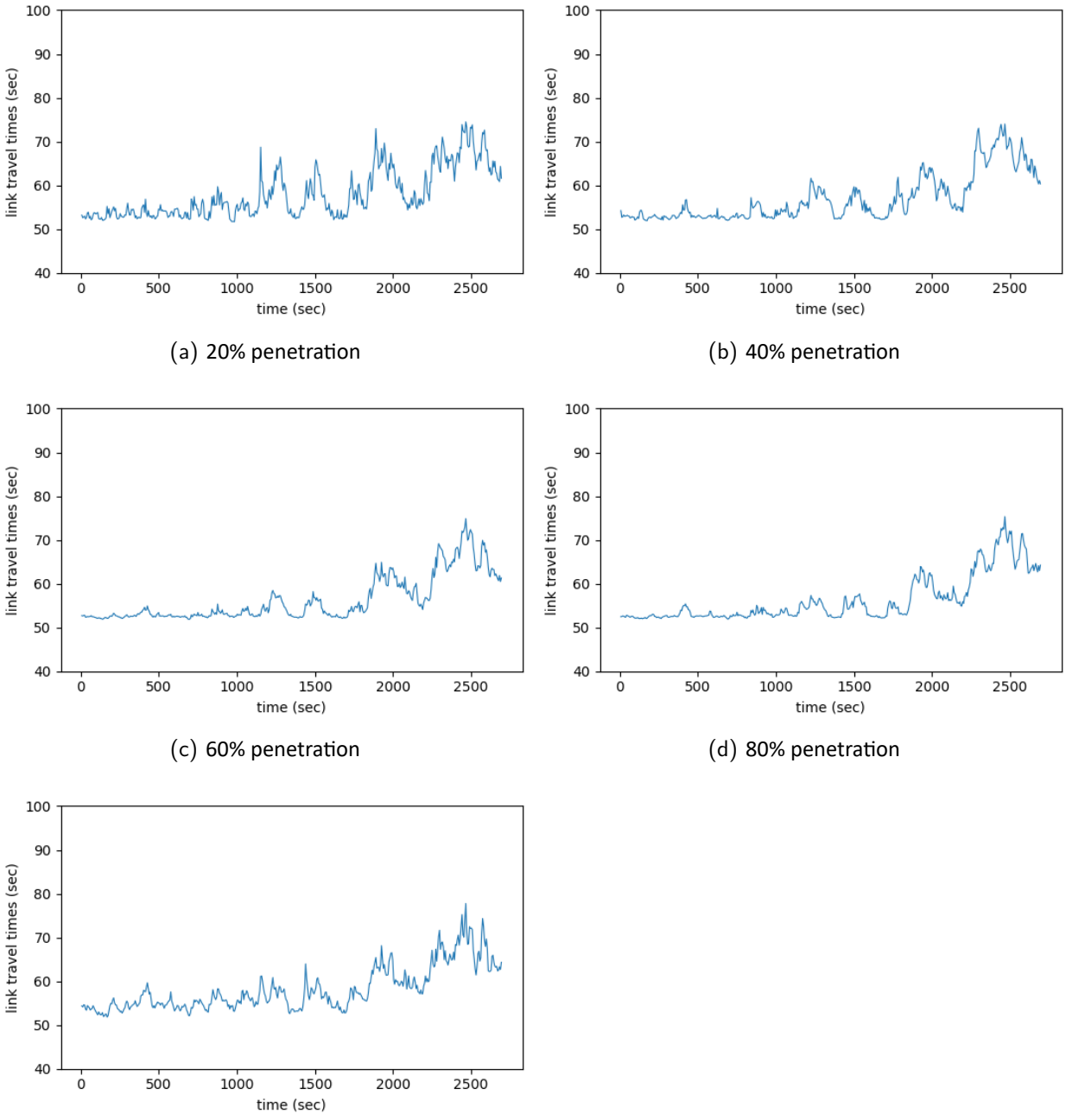
(b) Occupancy error

(c) Speed error

(d) Flow error

Figure. 18: Estimation errors under different penetration rates with BSMs from microsimulation data (uncongested condition)

Figure 18 shows the overall error or the error by cells (shown in equation (31)) for the experiment under an uncongested condition. As we can directly get the measurement of vehicle speed, the speed estimation has the highest accuracy while the density estimation has the lowest accuracy. With an increase in the penetration rate, the estimation accuracy increases. The last cell has the lowest estimation accuracy in density estimation while the first cell has the lowest accuracy in both speed estimation and flow estimation. As the input flow to the first cell is unknown, the only information that can be used to estimate the input flow is the number of detected vehicles entering the road section.



(a) 20% penetration

(b) 40% penetration

(c) 60% penetration

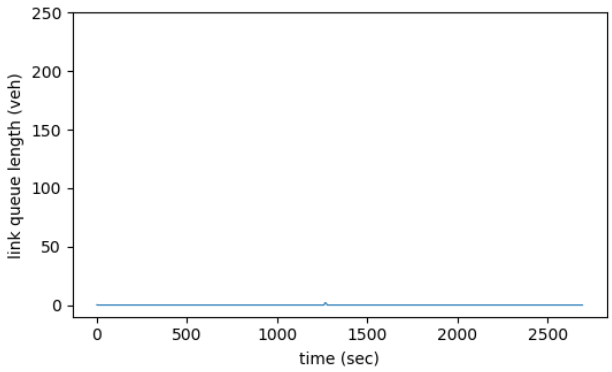
(d) 80% penetration

(e) 100% penetration

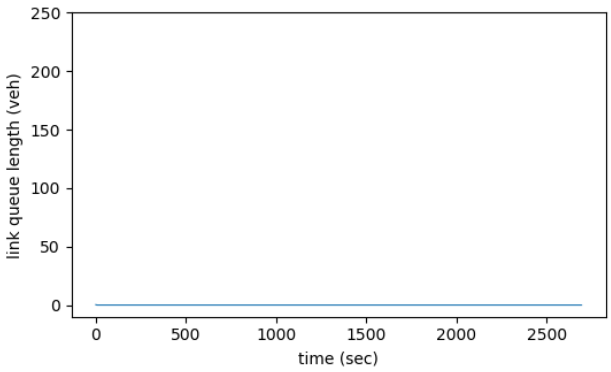
Figure. 19: Link travel time estimation under different penetration rates

The link travel times are estimated using the estimated speed. It is calculated by summing up travel times in all cells. The estimation of link travel times show high accuracy. Under all penetration rates, the travel time is 52 seconds at first, and then increases to about 70 seconds.

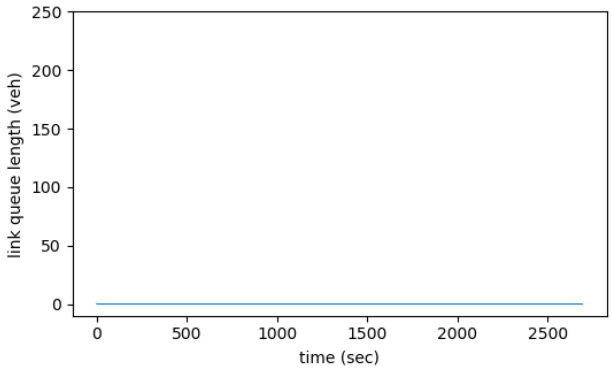
$$\text{Link Travel Time} = \sum_i \frac{L_i}{\hat{\bar{v}}_i} \tag{32}$$



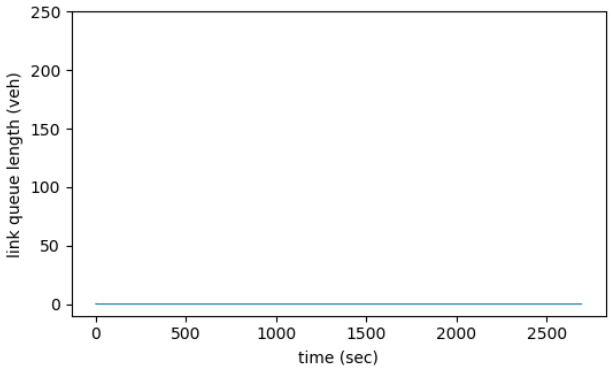
(a) 20% penetration



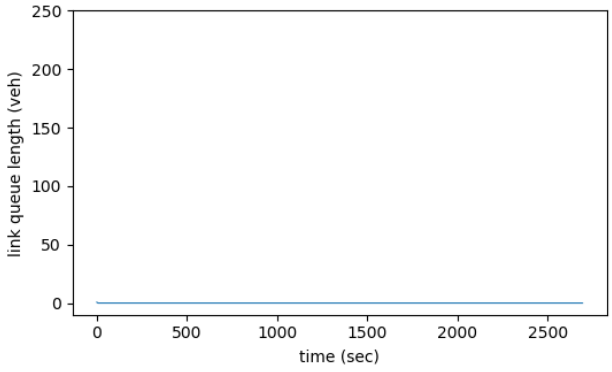
(b) 40% penetration



(c) 60% penetration



(d) 80% penetration



(e) 100% penetration

Figure. 20: Link travel time estimation under different penetration rates

We also estimate the queue length and the calculation algorithm is shown in 1. When the road is congested, a queue may form at the downstream of the road section and dissipate to the upstream of the road. The queue length is estimated by adding up the number of vehicles in each cell from the last cell to the first cell of the road. It is assumed that only if the last cell is congested, there is a queue formed and will spread to the upstream of the road. Otherwise, the queue length is 0. A cell is considered as a congested cell if its average speed is smaller than the critical speed v_c. However, in this scenario, there is no congestion on the road, so the queue length is 0.

Algorithm 1 Queue length calculation

The total cell number is i, the queue length q is 0 vehicle
for every cell in a link (start from downstream to upstream) **do**
  if $\hat{v}_i \leq \hat{v}_c$ **then**
    $q = q + \hat{n}_i$
  else
    Stop the queue length calculation
  end if
end for

6.4 Congested scenario

Figures 21 and 22 shows the BSMs generated from the simulation under congested condition with different penetration rates. As shown in these two figures, the congestion shows after about 25 minutes of the simulation, then the average travel time starts to decrease and the density starts to increase. Lane 1 has the most severe congestion among three lanes.

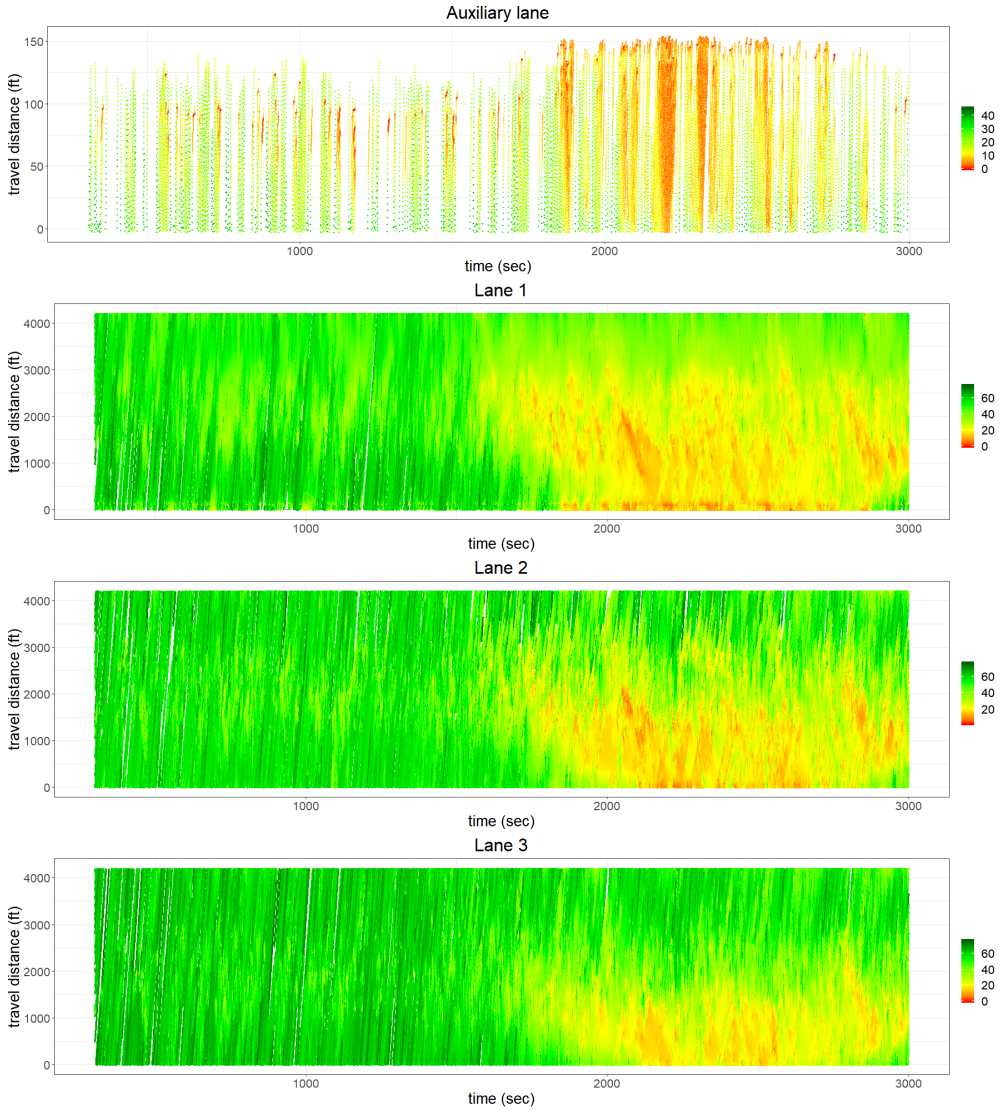
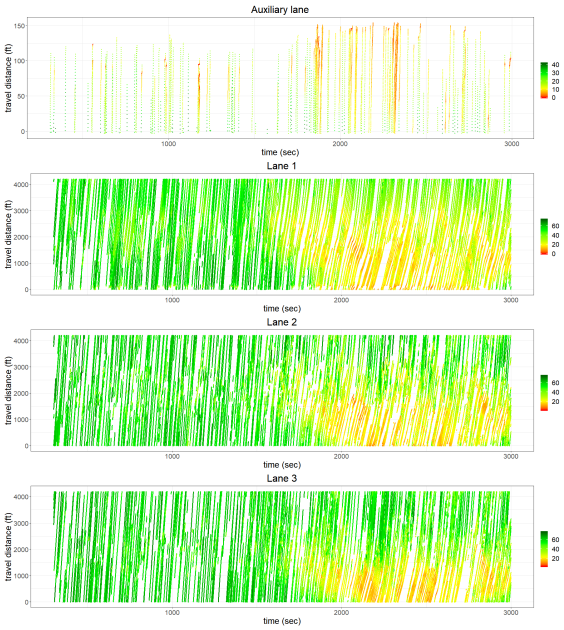
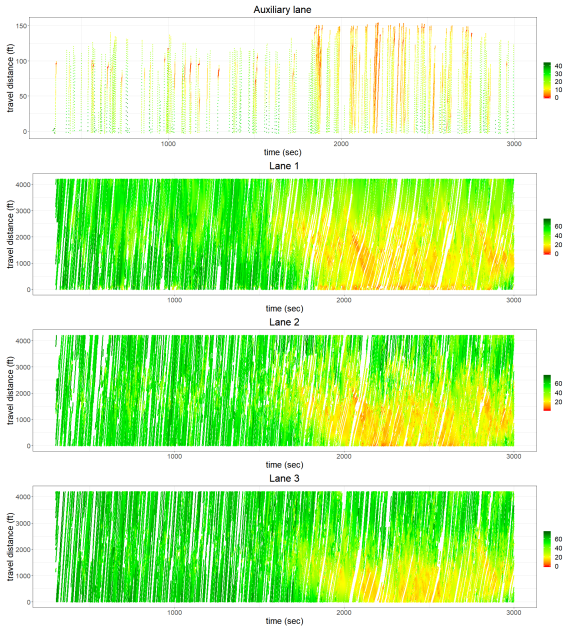


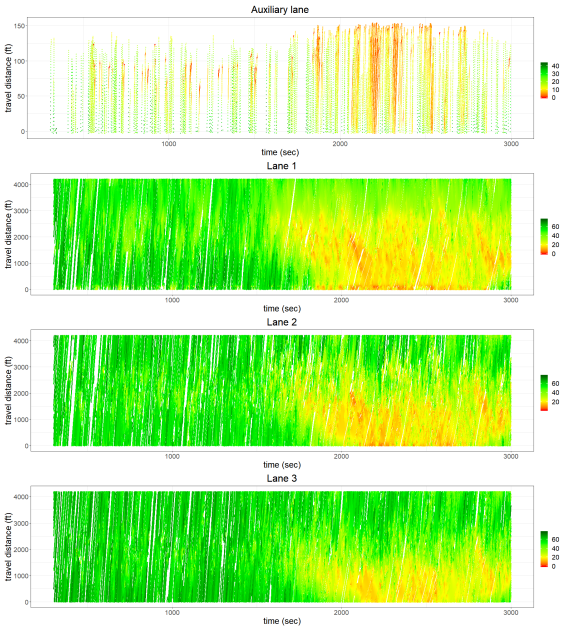
Figure. 21: 100% BSMs from microscopic simulation model (congested condition)



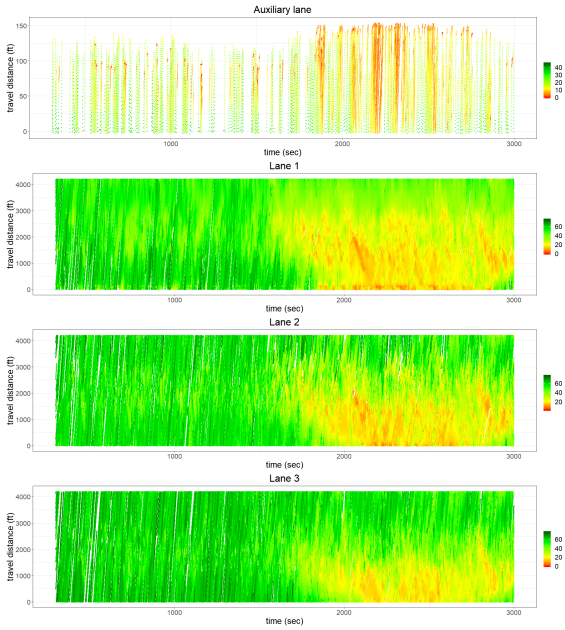
(a) 20% penetration



(b) 40% penetration



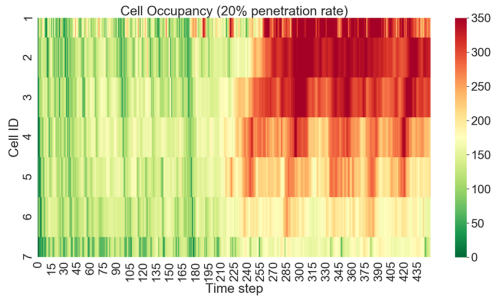
(c) 60% penetration



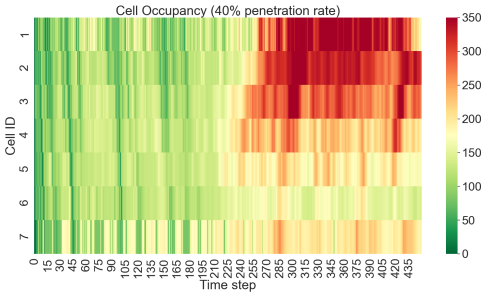
(d) 80% penetration

Figure. 22: BSMs from microscopic simulation model (congested condition)

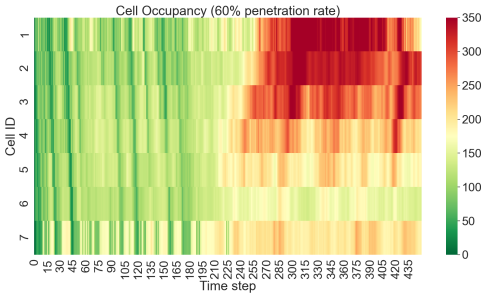
6.4.1 Traffic state estimation results



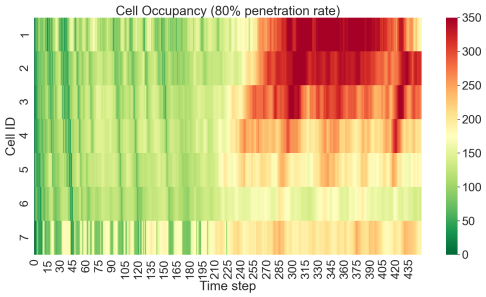
(a) 20% penetration



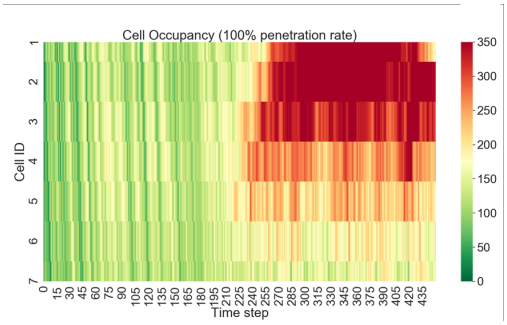
(b) 40% penetration



(c) 60% penetration

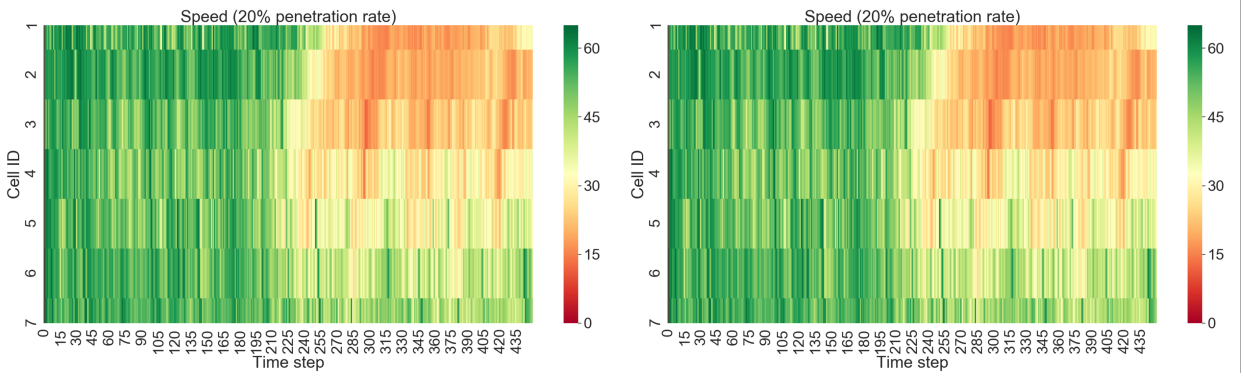


(d) 80% penetration



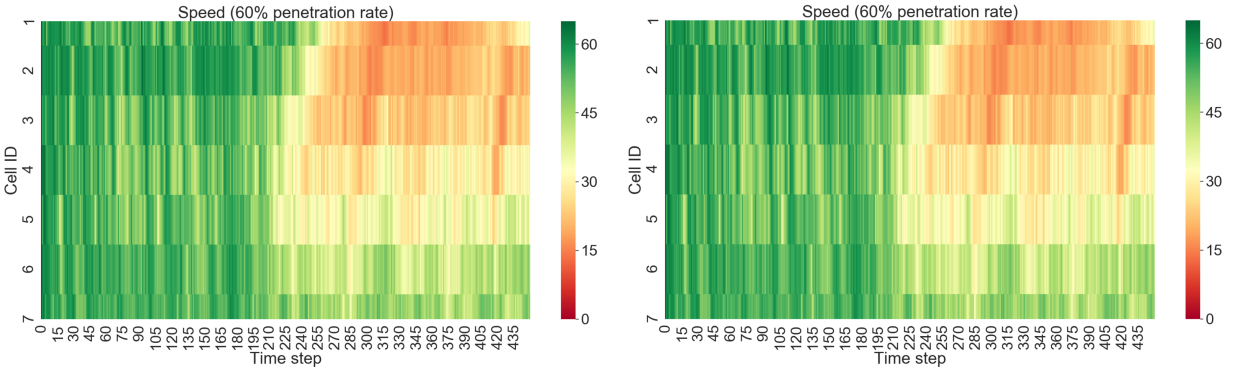
(e) 100% penetration

Figure. 23: Density estimation under different penetration rates with BSMs from microsimulation data (congested condition)



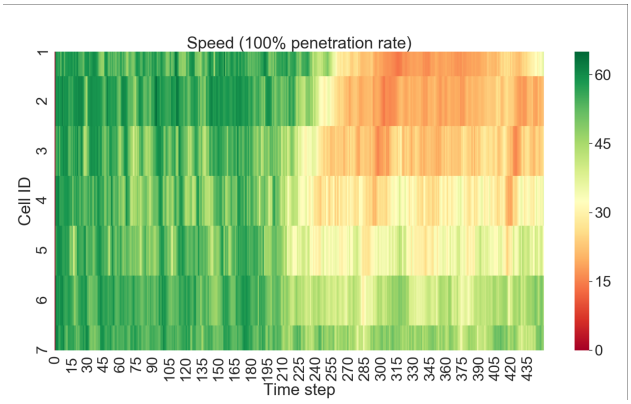
(a) 20% penetration

(b) 40% penetration



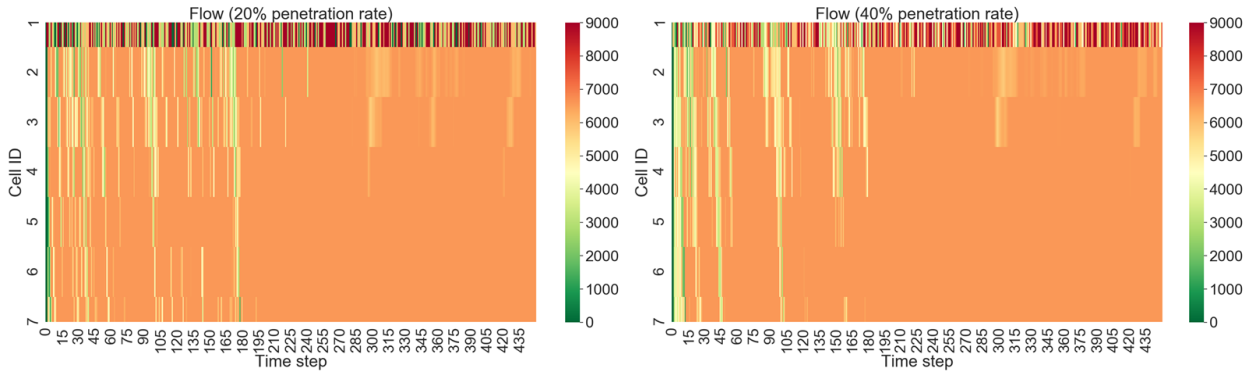
(c) 60% penetration

(d) 80% penetration

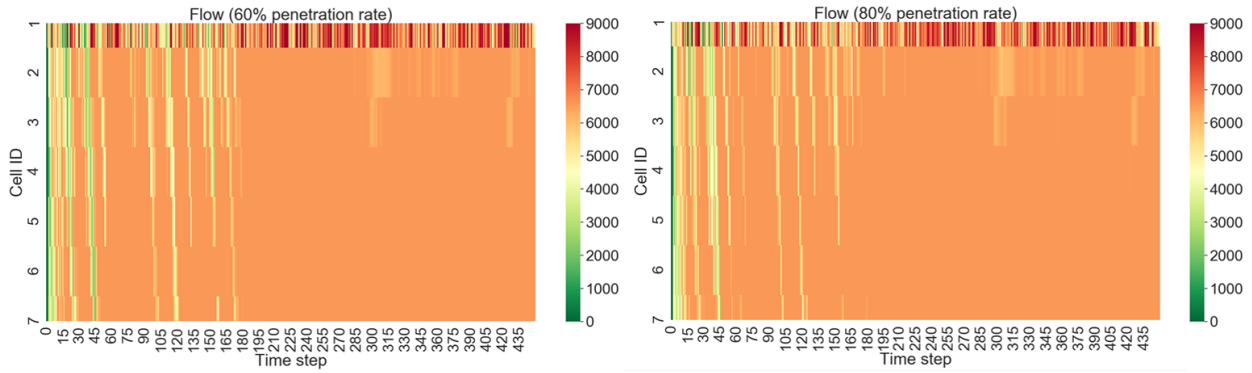


(e) 100% penetration

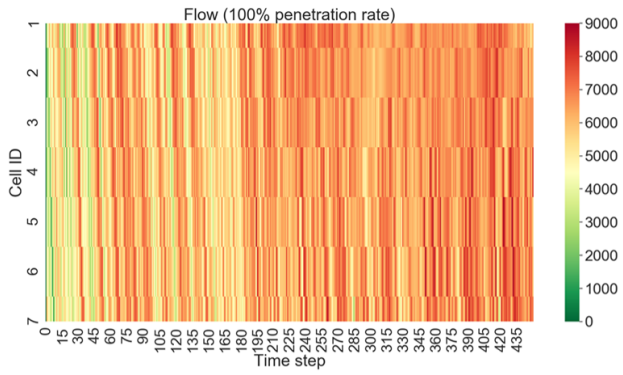
Figure. 24: Speed estimation under different penetration rates with BSMs from microsimulation data (congested condition)



(a) 20% penetration (b) 40% penetration



(c) 60% penetration (d) 80% penetration

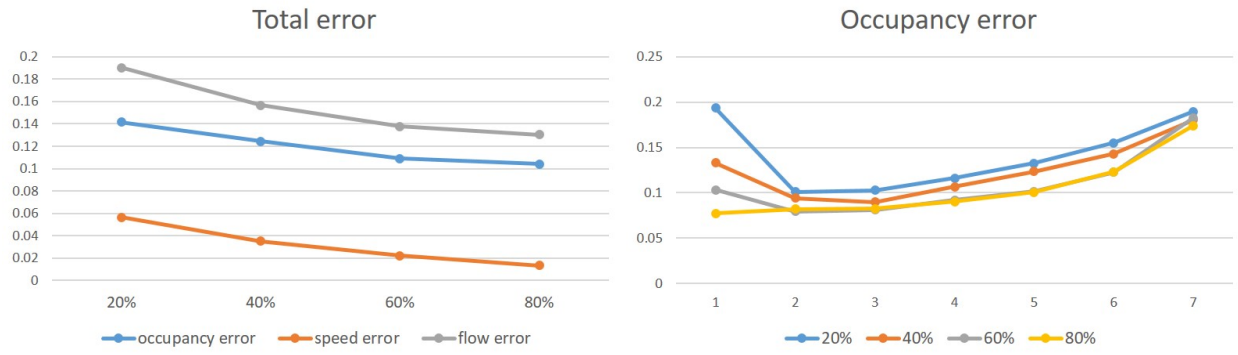


(e) 100% penetration

Figure. 25: Flow estimation under different penetration rates with BSMs from microsimulation data (congested condition)

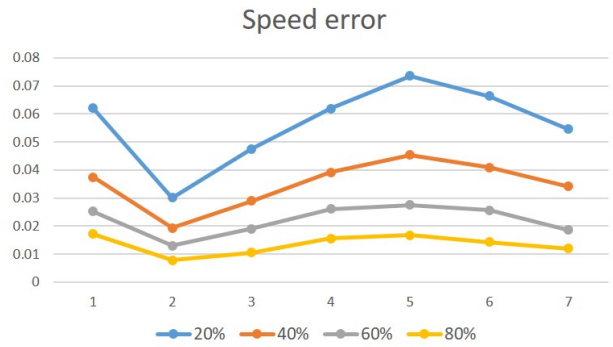
Figures 23, 24, and 25 show the traffic state estimation results compared with the actual traffic states. Compared with the uncongested scenario, the cell densities are larger as the cell color in

the heatmap of the density becomes red. There is a shockwave that starts at time step 216 on cell 5, and then it spreads upstream. There are 5 cells out of 8 cells getting congested after time step 260. The heatmap of the estimated densities shows high similarity to the heatmap of the actual densities under both low and high penetration rates. In the heatmap of the speed, we can also clearly find the formation and the propagation of the shockwave from cell 5 to upstream cells as there is an orange cloud that appears at time step 216. In the heatmap of estimated flow, we can see there is an area in which the cell color does not change. However, in the heatmap of actual flow, the cell color changes between light orange and dark orange back and force. It shows that the variance of actual flows is larger than the variance of estimated flows.

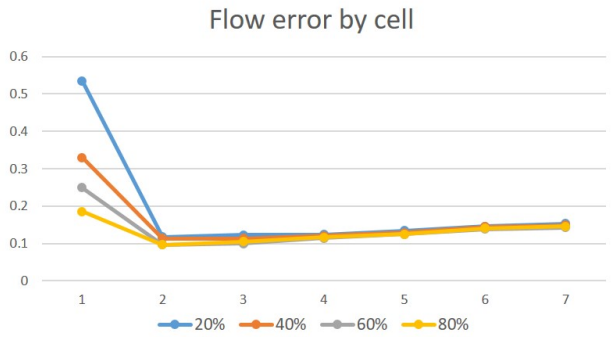


(a) Overall error

(b) Occupancy error



(c) Speed error

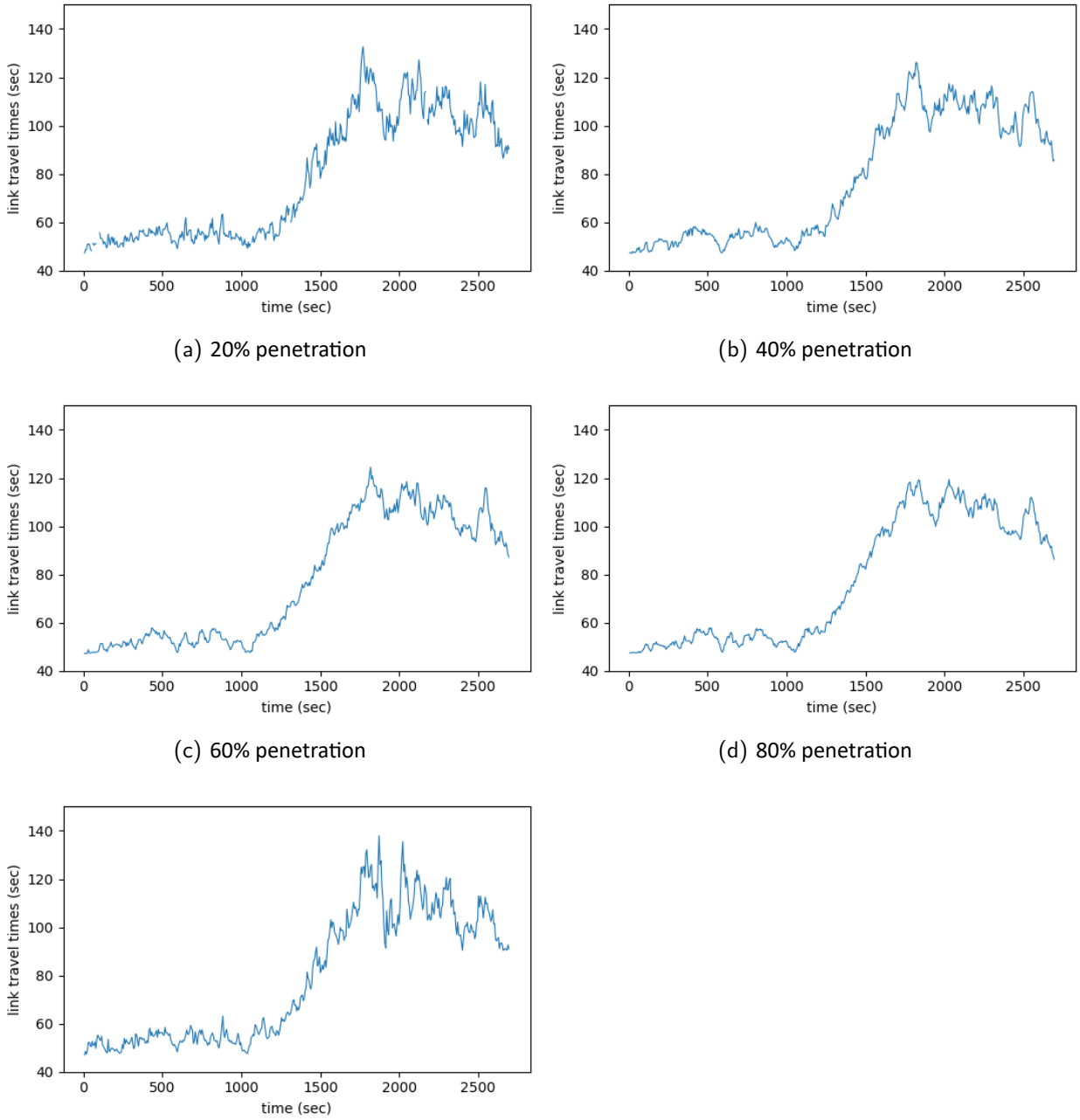


(d) Flow error

Figure. 26: Estimation errors under different penetration rates with BSMs from microsimulation data (congested condition)

Figure 26 shows the estimation error of the experiment under a congested condition. Overall, the accuracy of our model is much higher on a congested road. The flow estimation has the largest error whose maximum error is 20% and the speed estimation has the smallest error whose maximum error is 6%. An increase in the penetration rate increases the estimation accuracy. When

analyzing the estimation error by cells, two cells at two ends of the road section have the largest error in density estimation. The 5th cell in the road section has the largest error in speed estimation, but all speed errors are smaller than 7.3% and there is no big difference in the speed error between cells. For flow estimation, the first cell has the largest error due to the lack of enough information for input flows.



(a) 20% penetration

(b) 40% penetration

(c) 60% penetration

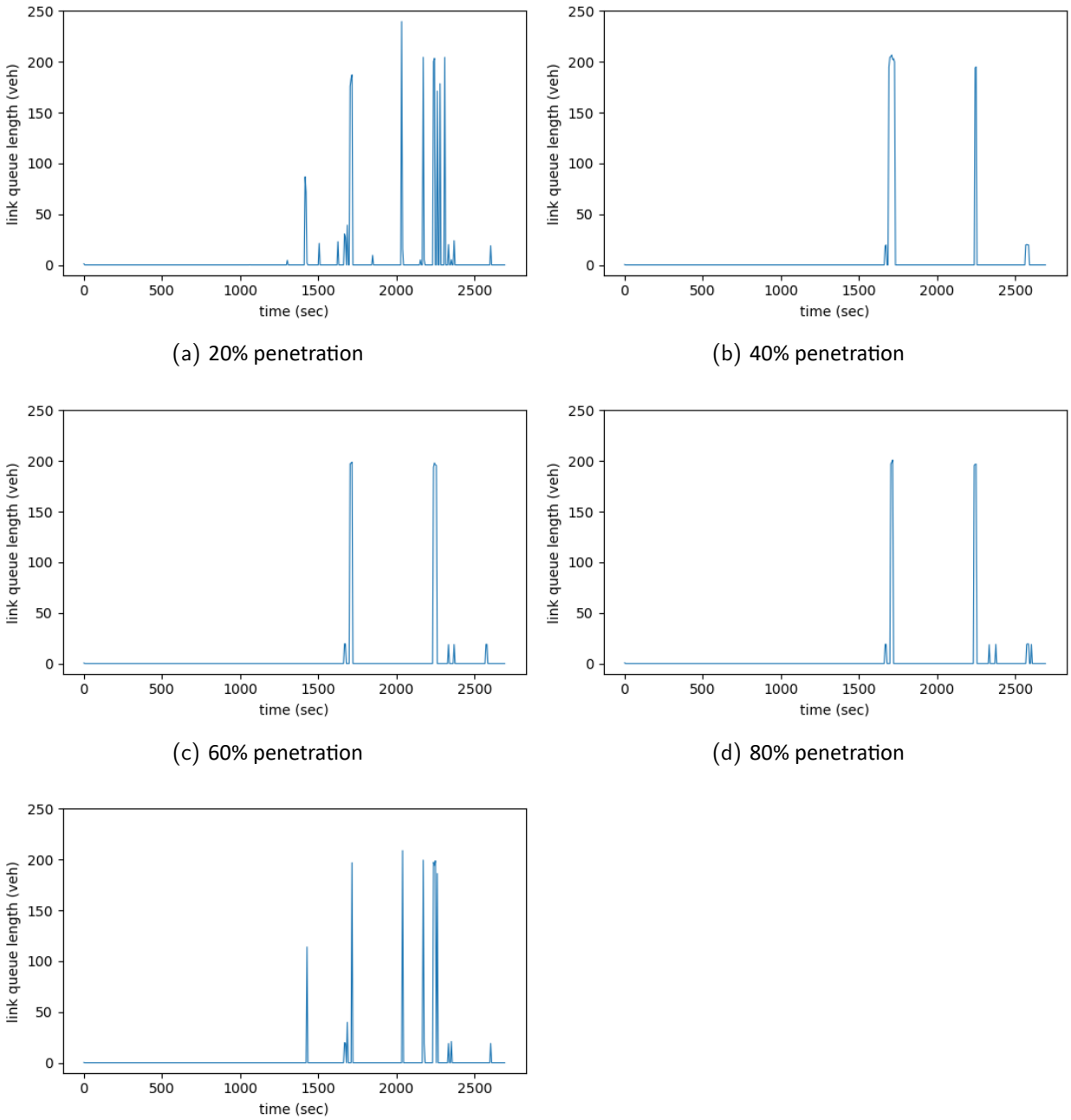
(d) 80% penetration

(e) 100% penetration

Figure. 27: Link travel time estimation under different penetration rates

The link travel times are estimated, as shown in Figure 27. It starts from 50 seconds and then increases to more than 120 seconds before it drops to 90 seconds. The estimation of link travel times under all penetration rate shows high accuracy. Figure 27 shows the estimation of queue length. The time when congestion appears and the queue length can be accurately estimated.

The estimation with a penetration rate of 20% tends to overestimate the queue length.



(a) 20% penetration

(b) 40% penetration

(c) 60% penetration

(d) 80% penetration

(e) 100% penetration

Figure. 28: Link queue length estimation under different penetration rates

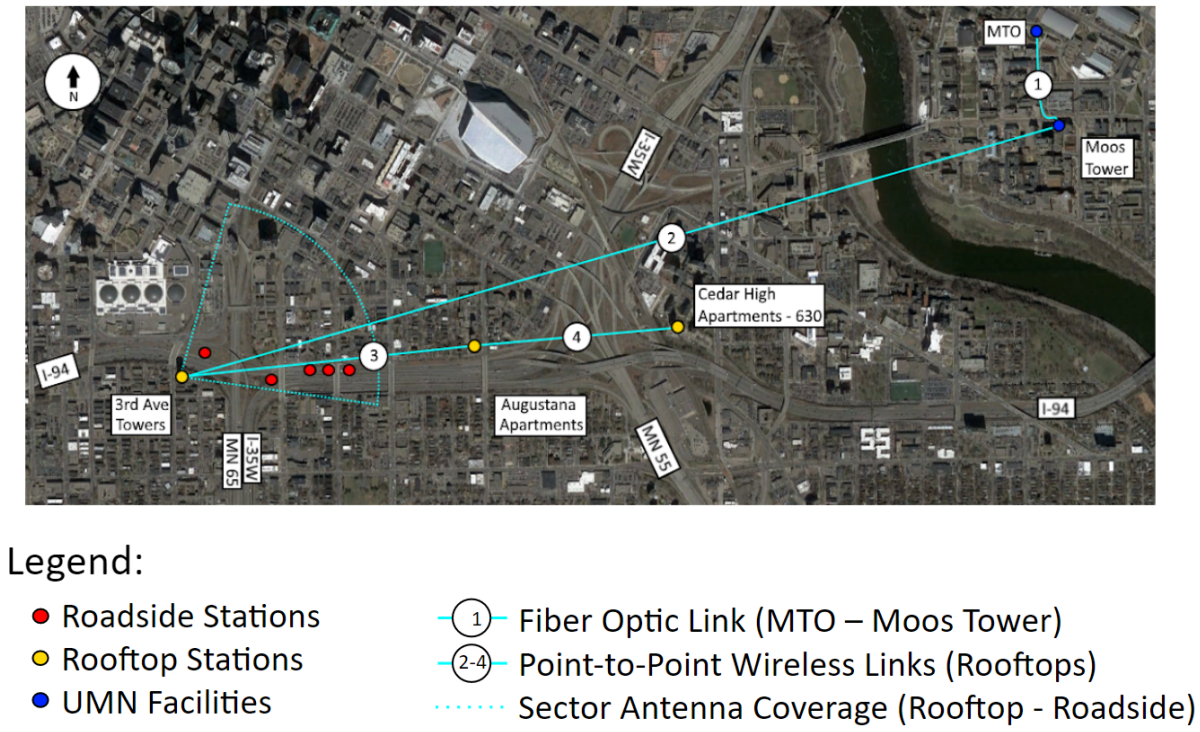


Figure. 29: Field data retrieval communication system

7 Real-world Vehicle Trajectory Collection

This task utilized the existing CV Testbed of Minnesota Traffic Observatory to provide real-world ground truth datasets for emulating BSMs. To understand why extracting seamless vehicle trajectories and emulated BSMs was an involved task, the process of collecting data from the site, transmitting it to MTO for storage, and accessing the database will be discussed.

7.1 Data gathering

The Testbed is made up of seven 24Ghz radar and cameras along a half-mile stretch of I-94 westbound, and was built on an existing MTO research field lab. From the Testbed, data is transmitted along a wireless communication network back to the MTO, utilizing existing rooftop nodes put in place for the original field lab. Figure 29 below shows how the Roadside Stations are able to transmit data back to the MTO. Before discussing extraction and cleanup of the radar data into emulated BSMs, it is important to understand the system architecture of the MTO's database, which allows access to historical and near-real-time data critical to the development of software discussed below.

7.2 Database storage process

Once the data arrives back at the MTO, generally within one second, it is stored in a PostgreSQL database in real time. This database also contains all historical data from the sensors. The data is saved exactly as it is output by the sensor driver without modification to preserve the original state, allowing further processing to be done without changing the raw data. In addition to the raw sensor data, the database also contains historical and current position and orientation data for each of the sensors, providing an automatic means for combining data from multiple sensors into a single reference frame. Finally, the database contains lane definitions for the corridor in the reference form of the sensors, allowing target data to be placed into a lane for analysis. Because single sensor will produce several million target measurements in a single day, data from the sensors is split into individual tables for each date to reduce the maximum query time. Data is organized into three tables for each date to provide different levels of resolution to analyze the data. At the highest resolution available, in the "trajectories_<date>" table, each record consists of a single target measurement, with an X and Y position and velocity value for each target at each point in time. One level further out, in the "objects_date" table, each record consists of a single target as viewed by one sensor over its entire life, containing information like the time of its first and last measurements, the estimated length of the target, and its average speed. Finally, in the "frames_date" table, information about the sensor frames that were decoded is provided, including the message count from the sensor and the number of targets observed during that instant in time (up to 64 unique targets). Together, these tables provide a complete picture of the data read from the sensor while reducing the amount of redundant information stored by the database. A diagram showing these tables and their relationships with each other is shown in Figure 30.

In addition to the data tables, there is another table containing both historical and current sensor position and orientation information. In this table, each record contains the sensor ID, the latitude, and longitude of the sensor, the X and Y coordinates of the sensor relative to the first (most-downstream) sensor in the network, and the start and end time for which this configuration is valid. For the current configuration, the end time is left blank to indicate that it is still in place. Additionally, the direction of the sensor (upstream or downstream) is also included, since it is necessary to understand the coordinate system of the sensor relative to the others; while the sensors contain their own orientation configuration that is used to translate ranges and angles into XY coordinates internally, the direction relative to moving traffic is still needed. The table containing lane information uses an integer ID for each lane, with lane 0 starting at the right-most edge of the road. Lanes are also given a name and designated as either a main or auxiliary lane, as one of the lanes in the corridor exits midway through the sensor installation. The geometry of

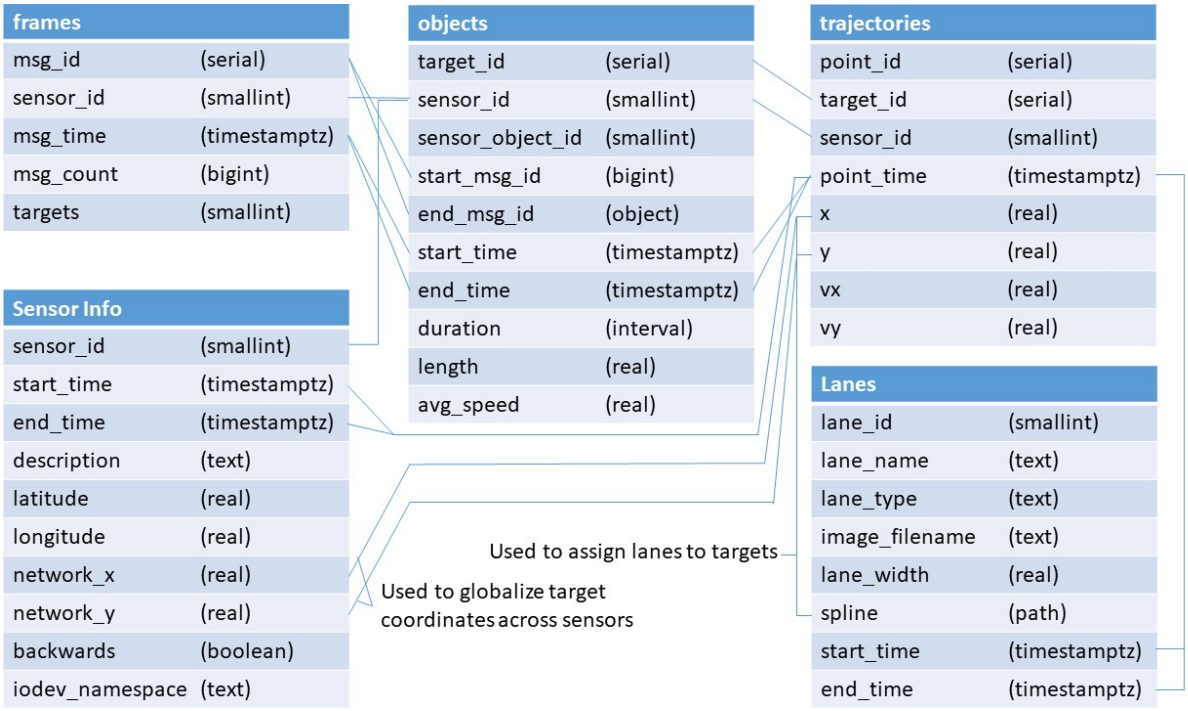


Figure. 30: Database diagram depicting schema of trajectory data from sensors.

the lanes is indicated by a spline representing one edge of the lane along with a lane width. This allows targets to be sorted into a lane at each point in time by measuring their distance from the spline and determining if it is within the lane, based on the lane width. Lanes are also defined for a period of time, which allows the system to accommodate changes in road geometry that would occur due to construction. This configuration is utilized by the visualization tools developed during this project. To simplify the use of the sensor position/orientation table and lane information table for adding information to raw sensor data, a number of SQL functions are also included in the database. These functions use the sensor positions/orientations to transform target coordinates into a global reference frame that is uniform across the network of sensors and use the lane information to assign a lane to each target at each point in time based on that translated position. This allows users to merge the data from multiple sensors and add this lane information in a single query without having to worry about the details of where the sensors or lanes were located at any particular point in time.

7.2.1 Inter-Process Communication for Exchanging Data

In addition to the PostgreSQL database, data is routed through a propriety Inter-Process Communication (IPC) system, meant to decode and share sensor data for time-critical safety applications.

Table 6: Sensor data fields available in IPC data

Field Name	Data Type	Description
time	Double-precision float	Unix timestamp with microsecond resolution
count	Unsigned integer	Cycle count for the radar
status	Integer	Status indicator
targets	Integer	Number of targets in message (up to 64)
id[64]	(Array) Unsigned short integer	Array to hold ID of each target
x[64]	(Array) Float	Array to hold X position of each target
y[64]	(Array) Float	Array to hold Y position of each target
vx[64]	(Array) Float	Array to hold X component of velocity of each target
vy[64]	(Array) Float	Array to hold Y component of velocity of each target
relays[16]	(Array) Boolean	Array to hold status of relay triggers

The data fields used for radar data, along with their data types, are listed in Table 6. These fields are used by software applications to produce visualizations of the data.

By abstracting sensor data in this way, the IPC system facilitates quicker development of applications using the data, allows multiple applications to use that data without the potential to interfere with one another, and provides a number of tools to extend the functionality of the system without requiring additional development. While the driver application itself only runs in real-time, making applications responsible for buffering data themselves, the IPC framework comes with utilities for saving data into a binary file and replaying it later. This feature is useful for development, testing, and debugging in that it allows applications to use sensor data without requiring an actual sensor thereby allowing specific conditions to be recreated without complex testing setups. Figure 31 below summarizes the system architecture as a flowchart, for reference.

An IPC visualizer exists, but was not part of the emulated BSMs effort. All tools that are used to determine periods of adequate congestion while all sensors were active, and extract and interpolate seamless vehicle trajectories draw from the main PostgreSQL database.

7.3 Data extraction and clean up

The CV Testbed captures and saves vehicle trajectories in a raw, sensor-by-sensor format. This was adequate for the scope of the original Testbed project; however, to emulate BSMs the production of seamless trajectories, filling gaps in the records due to sensing issues, and elimination of duplicate and erroneous data are required. The MTO has developed several different tools to interact with the database, to visualize near real-time data as well as historical datasets. The first applications that were developed to demonstrate the functionality were visualizers. These are

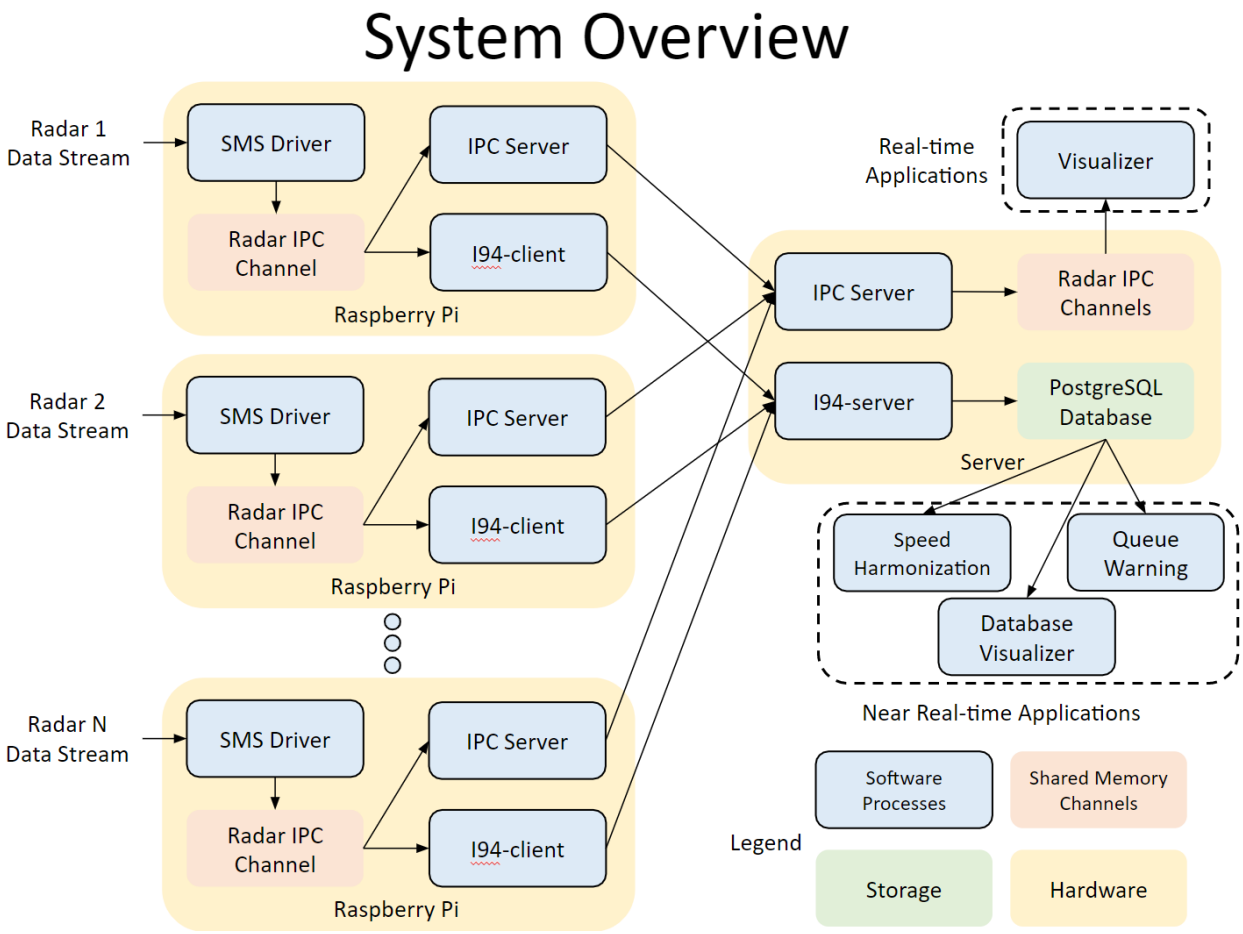


Figure. 31: Data collection system architecture

graphical programs ranging in complexity that display data from multiple sensors in a plan view, allowing users to see the data that is collected from the radar in an intuitive way.

7.3.1 Database Visualizers

The database visualizers require a configuration file containing sensor positions and orientations, but this file contains the same information used to adjust target positions in the database and can also be used to update this information in the database using a simple tool. This configuration file also contains a lane delineation algorithm. Currently, only the existing configuration file has been implemented for use in visualizing the raw data. The configuration file and current lane assignment algorithm does result in an imperfect visualization at points, where vehicles are duplicated or dropped in the programs, or assigned to the wrong lane. Examples of these issues will be shown below. Future configurations and algorithms to interpret the raw data can be implemented to reduce these issues, once developed. To extract and emulate trajectories and BSMs, three new

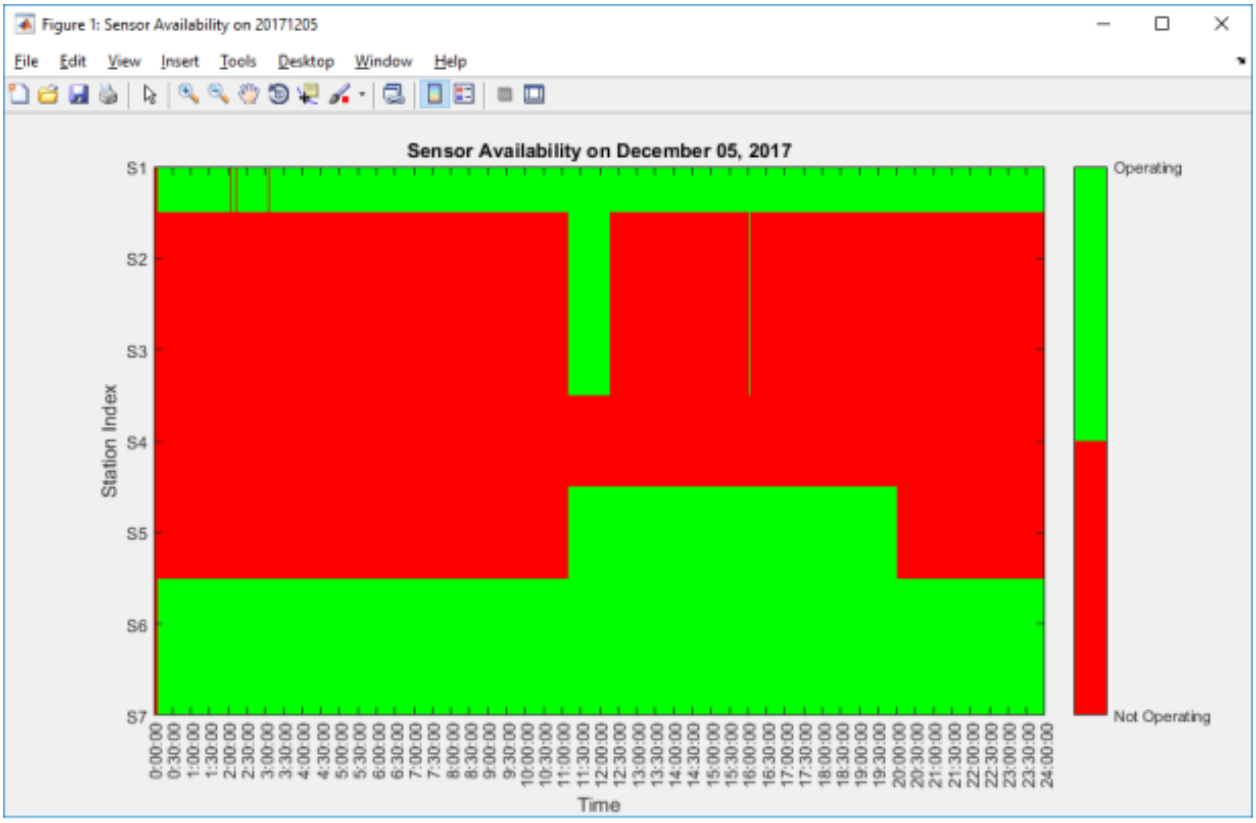


Figure. 32: Sensor availability heatmap tool

tools were developed to: see what sensors were active at any given time, extract and visualize vehicle trajectories both in Matlab and in Excel, and create time-space diagrams of vehicle travel along the Testbed. It is critical to note that these tools are based on the original alignment of the Testbed (ie, before any of the disassembly or movement of stations required by the 2017-present construction on I-94). New configuration files will allow for the new geometry of the Testbed to be displayed in the visualizers, in the future.

The first tool is a Matlab-based application used to see what sensors were functioning at any point on a given day. The output is shown in Figure 32. The Y axis represents the seven sensors, and the X axis is time, which can be anywhere from a five-minute period to an entire day. The tool checks each sensor in the database on the desired day and time to see if it was recording data or not. The resulting heatmap will look something like Figure 32, where a sensor that was functioning is green and any downtime is red. This tool is used once a desired period of congestion is determined, to see if all sensors were active during that period.

This tool is limited to one vehicle ID at a time, but can be a powerful tool for analysis as it is importing the data in to Matlab for the user to access.

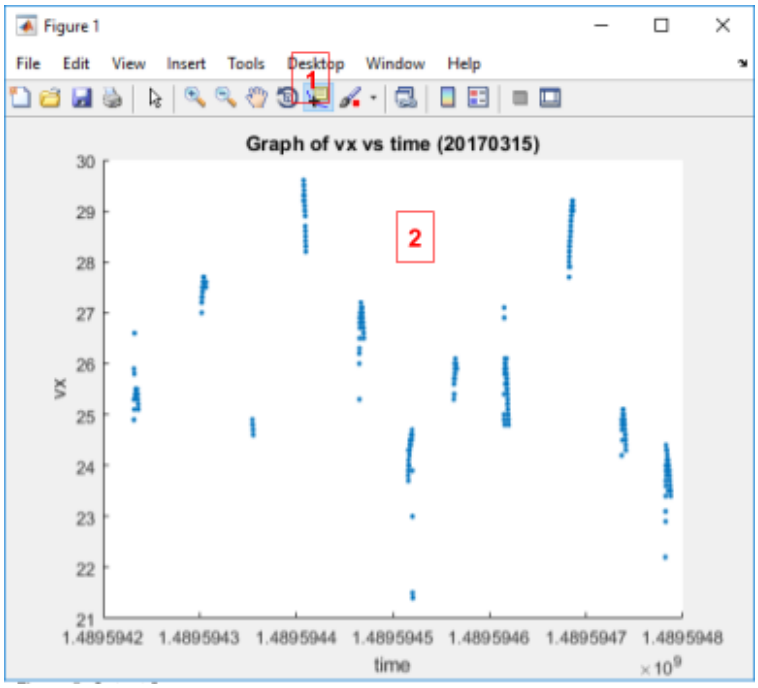


Figure. 33: A closer look at the output graph of vx vs time

The next tool is also Matlab based, and pulls vehicle location and velocity data from the database for Matlab to visualization and analysis. In a way, it functions as a User Interface for Matlab to communicate directly with the MTO's database. The user must choose which vehicle ID they would like to graph, as well as the time and day. Figure 33 shows an output of the program. Each vertical line represents a different vehicle's specific ID, which is limited by the radar's capacity to recognize up to 64 different objects. The limitation in recognition leads to vehicle IDs being recycled. It is one of several graphs the tool is programmed to produce, as the user desires. The time is shown in a UNIX timestamp. This tool is limited to one vehicle ID at a time, but can be a powerful tool for analysis as it is importing the data in to Matlab for the user to access.

The most complex of the tools, the Python-based Graphical Interface utilizes database visualization in a variety of ways. When users first boot up the program, they are able to choose whether to use the existing configuration file or custom configuration. As mentioned in the previous section, currently only the existing configuration file is available for use. Users will be taken to a home screen, as seen in Figure 34.

On this homescreen, users can see broad sensor availability for a month at a time. Clicking the desired day, users will then be able to see what periods of time all the sensors were available and select the time frame for the output time-space graph. Figure 35 shows this display.

After the user has selected the sensors and time frame they wish to see, they will click plot

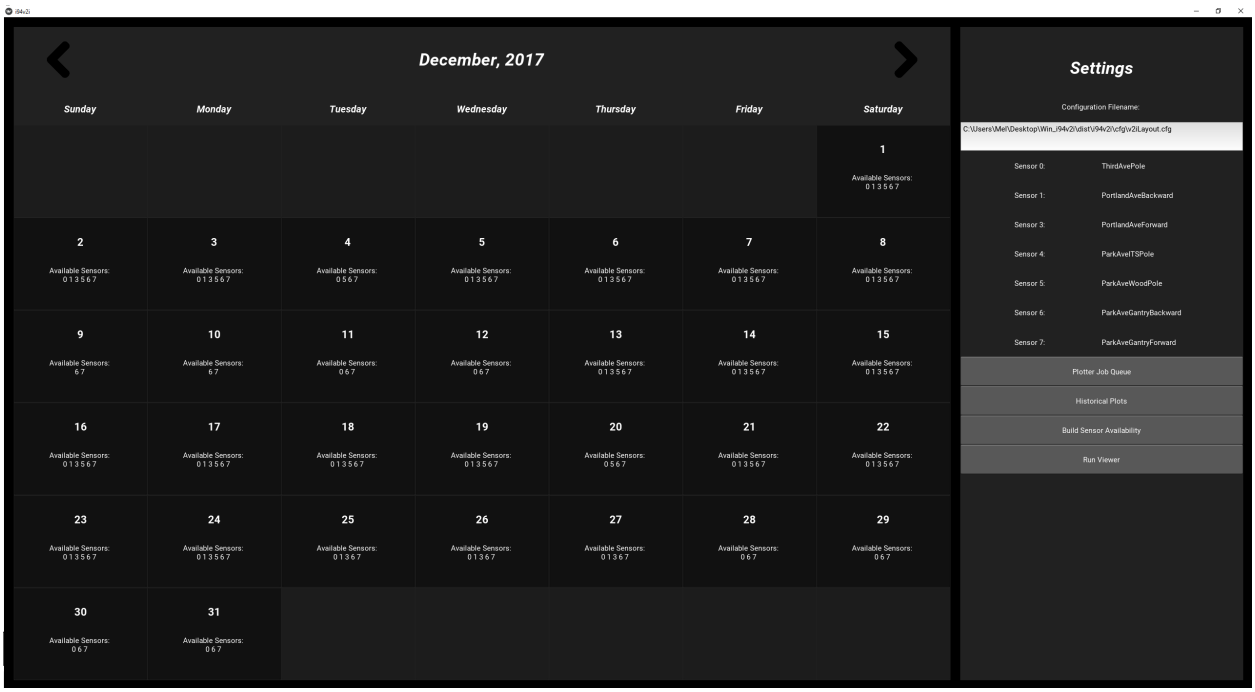


Figure. 34: Homescreen of the Graphical Interface tool

and wait for the program to run (this can take anywhere from a few minutes to hours depending on the amount of data being pulled from the database). On completion, the program will display a time-space diagram of all vehicle raw data for that time.

Figure 36 breaks down what sensor data is being displayed in each portion of the graph. Figure 37 shows the sections of radar coverage on a map overlay.

In Figure 36 and Figure 37, the vehicles are traveling from east to west and encountering the radar coverage in alphabetical order. Section A is data from the region between Chicago Ave and the gantry at Park Ave. Section B is from the gantry at Park Ave, covering the section underneath the Park Ave overpass. That is why the trajectories in section B of Figure 36 change color – clearly, it is the same vehicle with the same trajectory. Section B overlaps with Section C on the west side of the Park Ave overpass. For a limited time, targets are duplicated while the vehicles are picked up by two different sensors. Section C extends all the way to the Portland Ave overpass and overlaps with Section E. Section D is completely inside Section C, and a small part of Section E. There is a small blind spot at the west edge of the Portland Ave overpass, as Section E and Section F sensors are looking in opposite directions with no coverage directly between them. Section F is westward facing from Portland Ave, and section G is eastward facing from 3rd Ave. The break between F and G is due to the TH65 bridge, where the sensors were placed with as much line of site as possible but still could not pick up the space directly under the bridge.

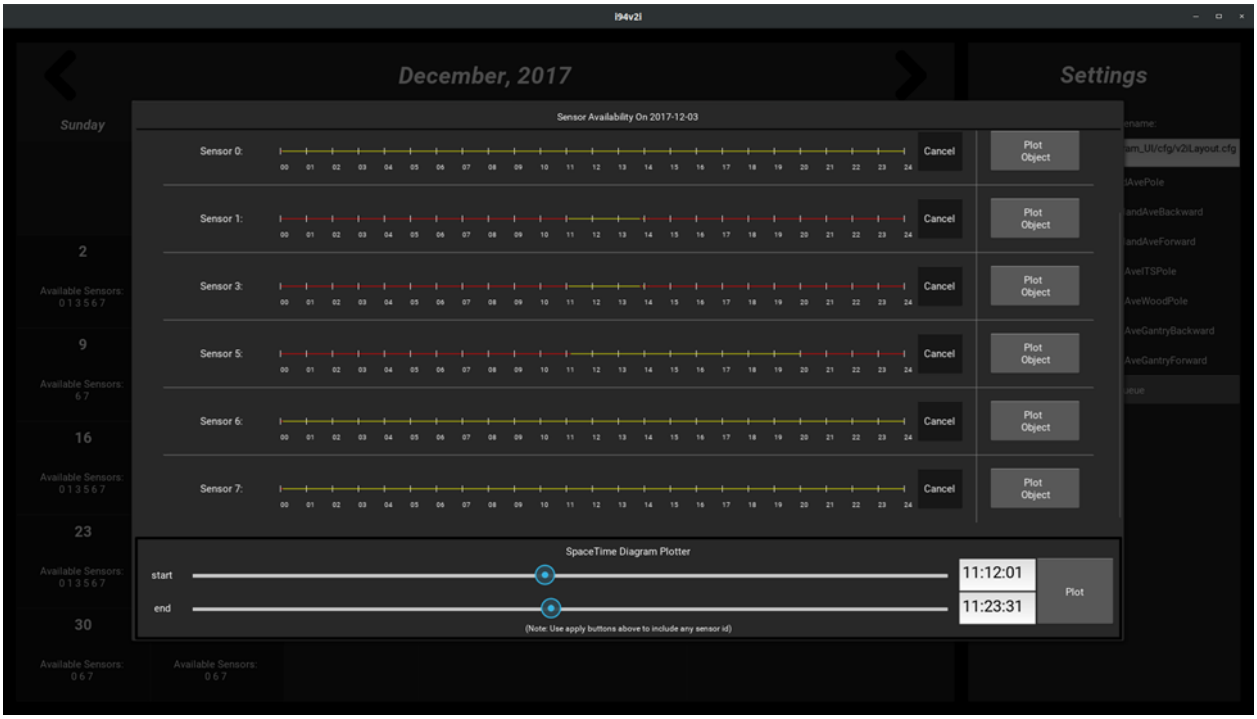


Figure. 35: Sensor availability per hour, and selection of time frame for output

In addition to overlap caused by radar placement, the configuration file and lane assignment algorithms sometimes misinterpreted data, leading to duplicate and misaligned vehicles. Figure 38 shows one such misalignment circled in red; vehicles are shown to have crossed paths, but no other signs of an incident appear. However, examining the slope of the misaligned vehicle reveals that it was on the left lane, and was placed into the right lane incorrectly by the lane assignment algorithm.

In addition to the misaligned vehicle, several duplicate trajectories can be pointed out. These result from radar overlap or sensor artifacts, the configuration algorithm inappropriately placing vehicle on the roadway, and/or the sensor position in the configuration file being off by a few feet. Figure 40 shows the same screenshot as Figure 38, but with duplicate trajectories circled.

Within the time-space diagram, users can also select specific trajectories that are of interest and export the raw data for only those trajectories into an Excel file for further analysis. Currently, no tool exists to extract vehicle trajectories with the same slope over the entire length of the testbed. To knit together continuous trajectories, they must be examined by inspection and selected for export. Figure 41 shows a zoomed in look at a single trajectory that has been selected for analysis. The color change indicates a different vehicle ID, but by inspection of the slope, it is the same vehicle passing by multiple sensors.

Table 7: Output of selected trajectory, showing time, vehicle ID, sensor ID, position and velocity

	timestam	count	object_id	sensor_id	x	y	vx	vy	v (magnit	length	global x	global y	latitude	longitude
0	1.51E+09	3.21E+08	32	7	104.45	-6.72	-26.9	0	26.9	3.2	606.9528	-58.7051	44.96601	-93.2634
1	1.51E+09	3.21E+08	32	7	102.91	-6.78	-26.9	0	26.9	4.4	605.4128	-58.7651	44.966	-93.2634
2	1.51E+09	3.21E+08	32	7	101.12	-6.53	-26.9	0	26.9	5	603.6228	-58.5151	44.96601	-93.2634
3	1.51E+09	3.21E+08	32	7	99.84	-6.21	-26.9	0	26.9	5	602.3428	-58.1951	44.96601	-93.2634
4	1.51E+09	3.21E+08	32	7	98.5	-5.95	-26.9	0	26.9	5	601.0028	-57.9351	44.96601	-93.2634
5	1.51E+09	3.21E+08	32	7	97.22	-5.76	-27	0	27	5	599.7228	-57.7451	44.96601	-93.2634

Table 7 shows an example of the output, where "object_ID" is the vehicle ID. This table contains position, velocity and trajectory information.

This raw radar data becomes the real-world ground truth for the methodologies being developed in this project. As this radar data is collected at 20 Hz, twice the frequency of BSMs, only every other line will be used during emulation. The BSMs will include the velocity (magnitude), the latitude and longitude, and heading of the vehicle. The heading will be calculated using the latitude and longitude from two consecutive data points.

7.4 Emulating BSMs

The previous chapter provided a summary overview of the I-94 CV Testbed infrastructure and data collection architecture. It also highlighted the major issues involved in the collection of actual vehicle trajectories through radar. As pointed out, the radar data are not perfect since they include several artifacts and noise generated by the sensor as well as issues with the data harvesting architecture. In the course of this task, which represented the first real-world, large scale implementation of the I-94 CV Testbed, a number of previously undiscovered issues and bugs in the architecture were discovered. Given that the actual data had been collected more than a year before, it was not possible to repeat the data collection so a considerable effort was spent in finding the issues, which involved the development of algorithms that could scan the 1.8TB of collected vehicle trajectories, detect the discrepancies and produce summary visualizations that allowed the research team to identify the source of each problem and develop correction algorithms. Each of the four scan and correction cycles necessary to bring the raw data into a stable quality required three weeks of processing time for the database server. Even after the data was cleaned up and all the discovered errors and noise removed, the daunting task of stitching the trajectory pieces captured by the different radar sensors and filling in the gaps with estimates that follow realistic vehicle kinematics remained. The research team, as part of Task 6, is working on a large scale data mining methodology; due to the very large delay in this task it was decided that the identification of same vehicle's individual radar traces was going to be done manually, assisted by additional purpose made tools. Figures 42 and 43 present the latest evolution of the

Time-Space diagram visualization tool that allows for manual identification and marking of single vehicle trajectory pieces.

7.4.1 Data processing methodology

Specifically, the current batch of data extracted for Task 1 covers a five-minute period between 13:55:00 and 14:00:00 on December 22nd, 2017. Figure 42 shows the time–space diagram of the captured trajectories on the right lane between Chicago Ave and 3rd Ave on I-94 westbound. This segment is approximately 650 meters long and is as explained earlier is covered by six radar stations. (Note: one radar was not used that day as the data provided by it overlapped with other radars. Removing it led to less noise in the existing data.) The manual trajectory extraction methodology can be summarized in the following steps:

1. Having selected an appropriate day and time period, the tool retrieves from the database all the individual radar target frames.

2. Having retrieved the raw data, the internal lane assignment algorithm assigns a lane to each individual data point. This assignment, as explained earlier, is not perfect but it is a necessary step in the production of the Time-Space visualization. The underlying data does not change though.

3. Having confirmed that the selected time period has reasonable data for the purposes of this research, the entire set is extracted and stored in a local database file.

4. The tool allows the manual selection of trajectory pieces that the user identifies as belonging to the same physical vehicle. This step required careful progress to avoid stitching together trajectory segments from different vehicles. Figure 43 shows two such trajectory groups selected (highlighted by the tool) and the data below it present the output of the program. In this example, the data output read:
   17: 821 843 894 1015 1056 1114 1199 1221
   18: 679 722 829 897 924 1027 1017 1088 1198

5. The number before the ":" represents the new ID of the combined trajectory trace while the list of numbers following it are the unique ID's of the individual radar traces that belong to it. Notice that there is a fair amount of empty spaces in the Time-Space diagram. The radar on the I-94 CV Testbed, being Doppler radar cannot recognize standing targets. If the speed of a tracked vehicle drops below 0.2 m/sec it becomes "invisible" to the radar

sensors. In addition, once the vehicle gains speed above 0.2 m/sec, it takes the radar 12 cycles (0.6 seconds) to reacquire the target and start tracking. The traffic on the right lane during the selected time period, as it was usual in that section of I-94 was congested and one can recognize the familiar patterns of shockwave disturbances traveling upstream.

6. The second output of the new tool is a text file that contains the aforementioned manually identified vehicle trajectory groups. The graphs in Figure 44 present an example of the extracted information from one such trajectory group. Notice that the trajectories include a lot of discontinuities as well as overlapping segments in the areas where more than one radar sensor tracks the actual vehicle. In addition, especially the cross lane location of the vehicle has several outliers, typical of data captured from a down-lane radar sensor. The next step involves the cleaning up of noisy and erroneous data and the imputation of missing ones.

7. For this step, another tool was developed, this time in Matlab. The Combined Trajectory Generation Tool, employees a series of outlier detection and filtering methodologies to detect and reduce the noise and erroneous data points in the combined trajectory. For example, to eliminate the outliers and smooth the data in the Y (cross-lane) dimension a three term Fourier series model is fitted in the data. The resulting model also allows for the estimation of missing trajectory points. The X (down lane) and Speed series are smoothed and imputed through the use of a piecewise cubic spline model.

8. Given that the original grouping was done manually, it is possible to have included trajectory segments that looked appropriate in the time-space diagram but do not belong to the same vehicle. For this reason the tool allows for the visualization of all combined trajectories so the user can identify discrepancies, make adjustments in the input file containing the groupings and repeat the process.

9. The final output from the process is a series of individual files containing the combined, imputed, and filtered trajectory of an individual physical vehicle. The final output includes Latitude, Longitude information and has an update interval of 10Hz to emulate the required information contained in the BSMs. The following table presents a sample from the file containing the output for the vehicle shown in the earlier figures.

In total for the selected five minute period on December 22nd, 2017 the resulting dataset includes 144 vehicle trajectories on the Left lane, 141 trajectories on the Middle lane, and 122 trajectories on the right lane. Please note that the lane designation is purely organizational since

Table 8: Example of emulated BSM output

Vehicle-ID	Time	Global_X	Global_Y	Speed	Latitude	Longitude
34	1513972578.28193000	643.1856	-59.8705	18.5667	44.9661	-93.2631
34	1513972578.38192990	640.3964	-59.8536	18.7000	44.9661	-93.2632
34	1513972578.48192980	637.2193	-59.8442	18.7000	44.9661	-93.2632
34	1513972578.58192970	635.3279	-59.8511	18.7000	44.9661	-93.2632
34	1513972578.68192960	633.4836	-59.8697	18.7000	44.9661	-93.2633
34	1513972578.78192950	631.6270	-59.9043	18.7000	44.9661	-93.2633

a lot of these vehicles performed several lane changes during the half a kilometer path. For example, vehicle 34 shown as an example on Figure 45, as can be observed from its Global_Y coordinate time series performs at least two lane changes.

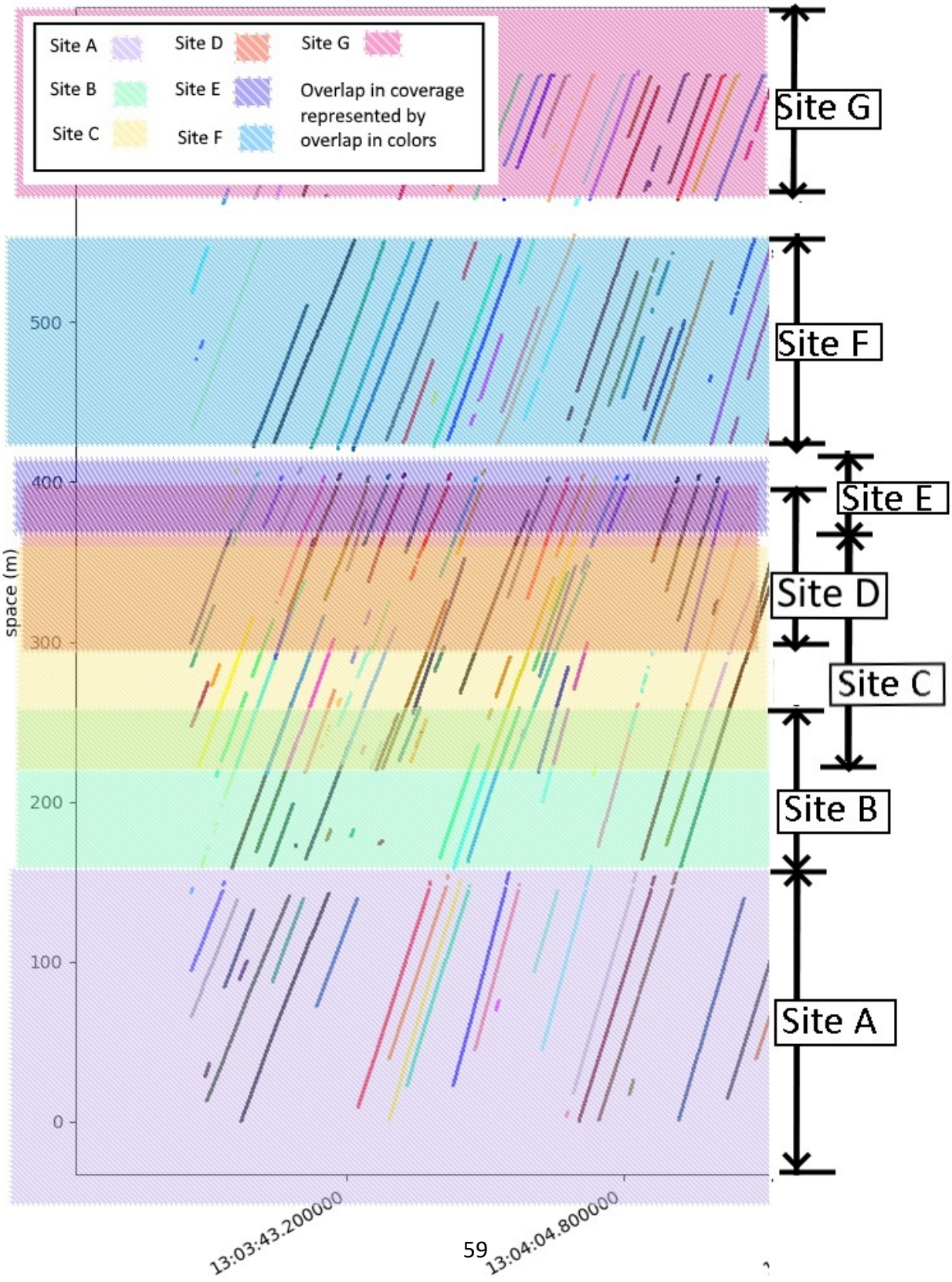
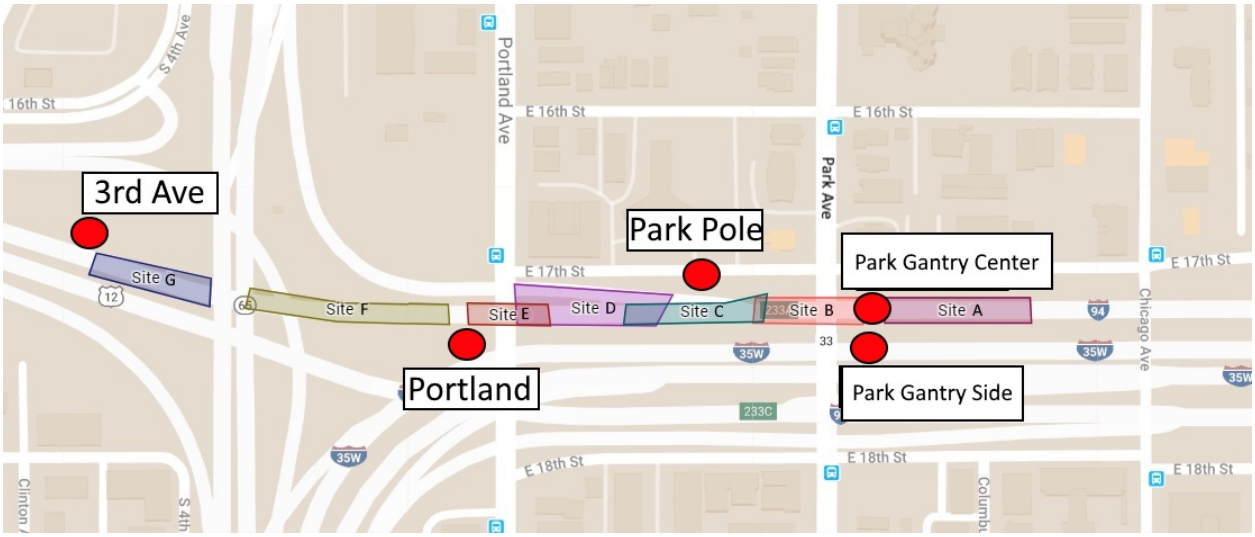


Figure. 36: Time-space broken down by sensor

Figure. 37: Radar (red dots) coverage areas (shaded), with overlaps



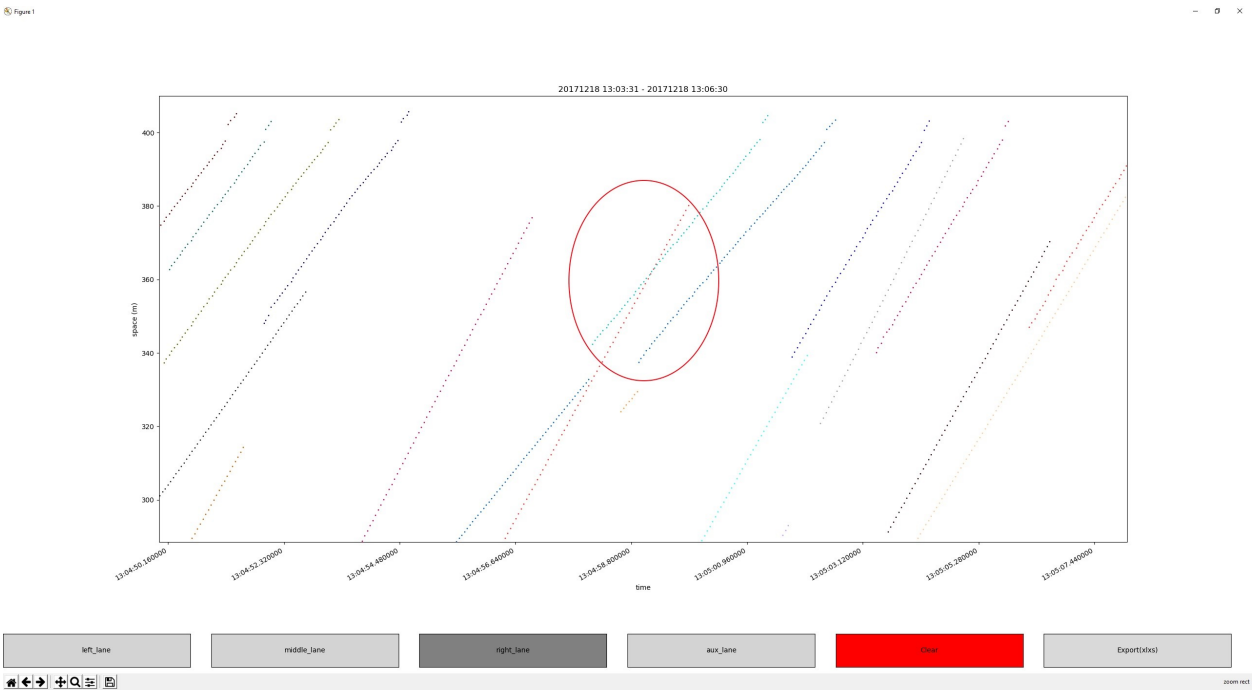


Figure. 38: Misaligned vehicle due to configuration file, on the right lane

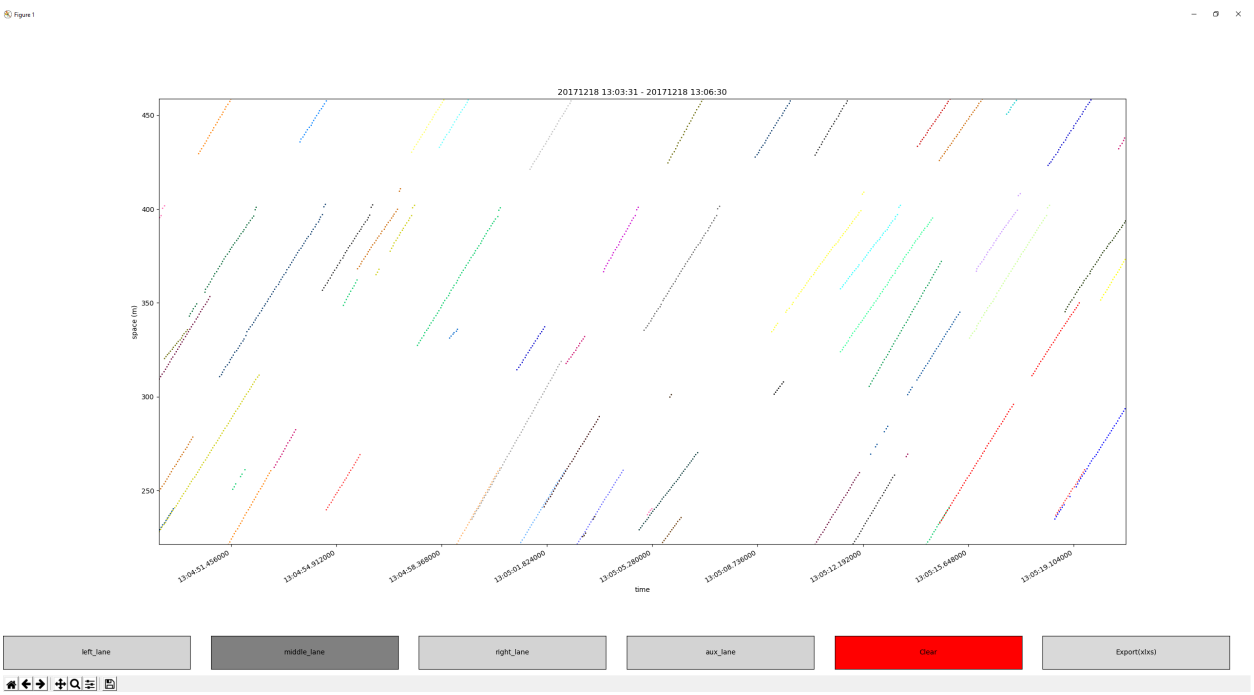


Figure. 39: Correct lane for the misaligned vehicle (left lane).

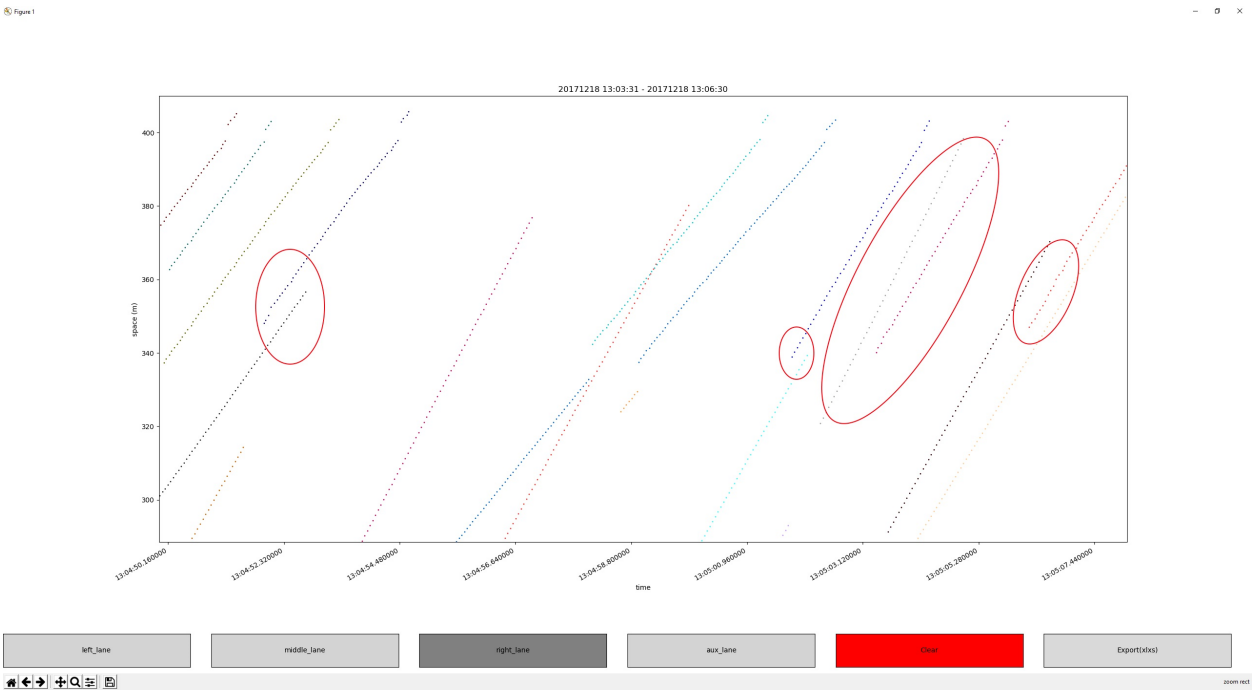


Figure. 40: Duplicate trajectories

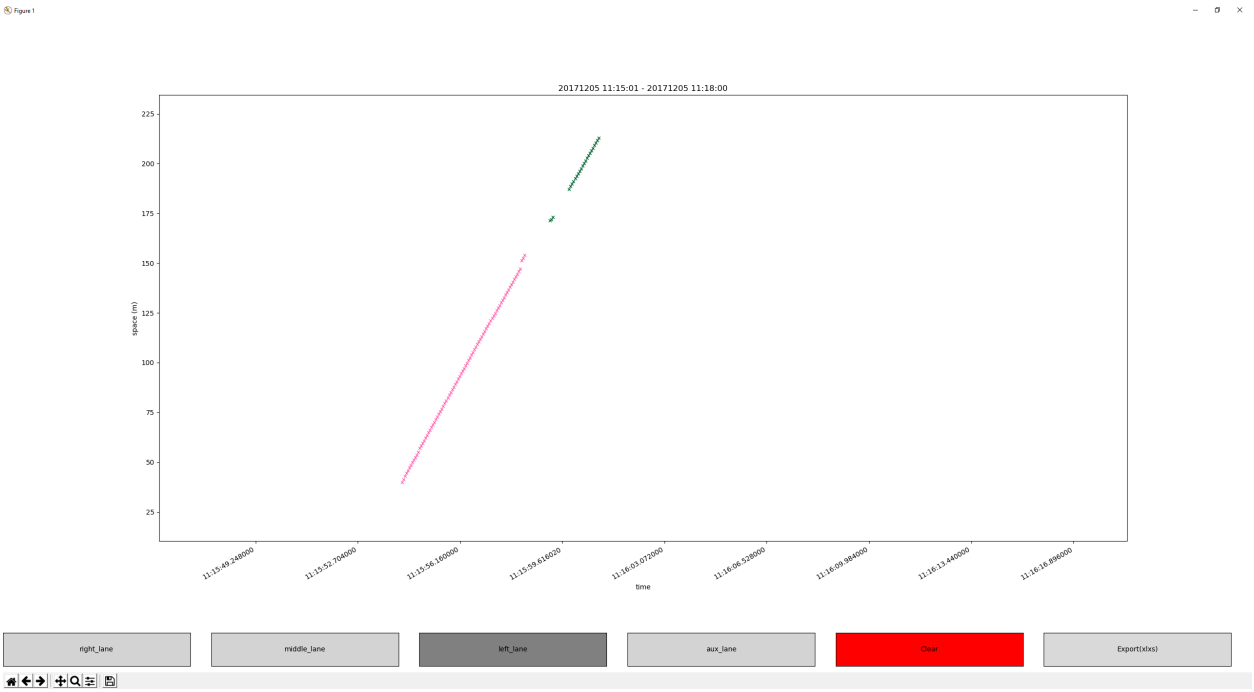


Figure. 41: Example selected trajectory. (icons have changed from dots to crosses)

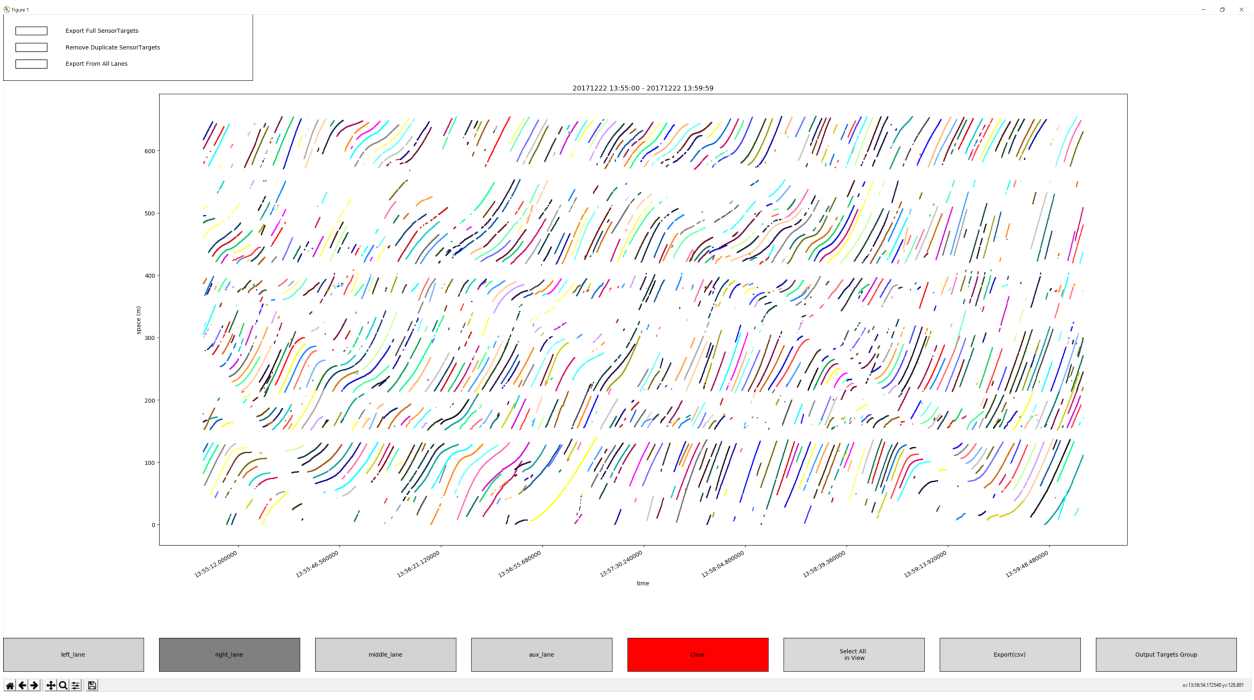


Figure. 42: Time – space diagram visualization tool

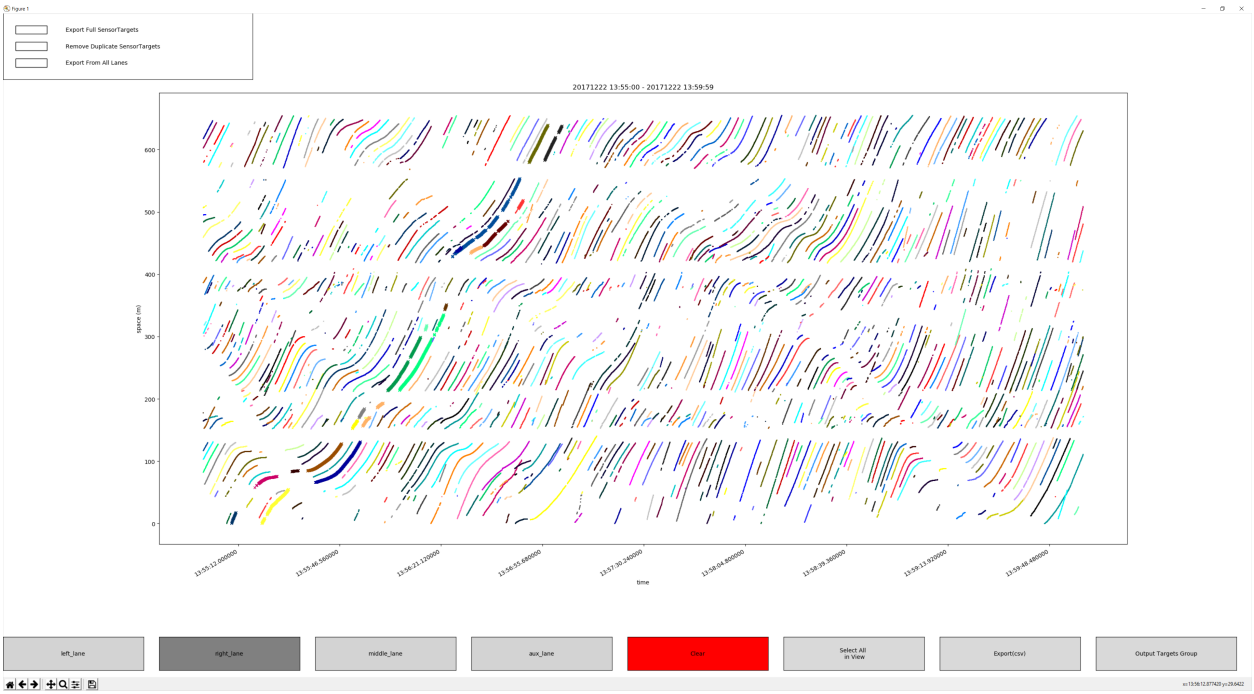
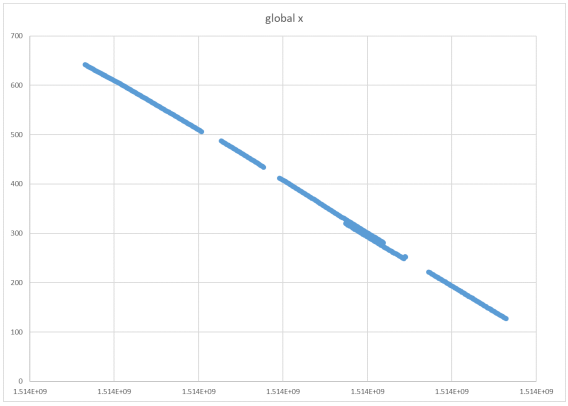
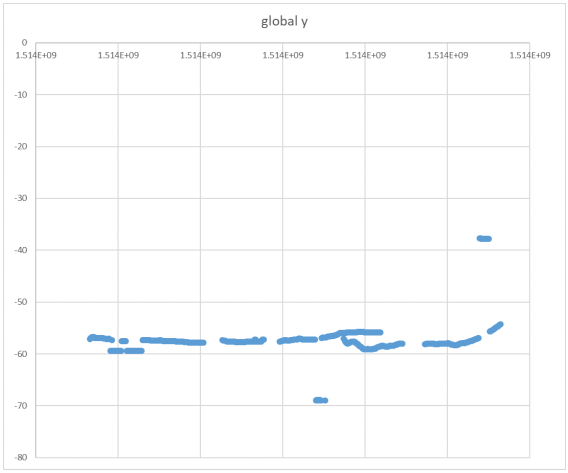


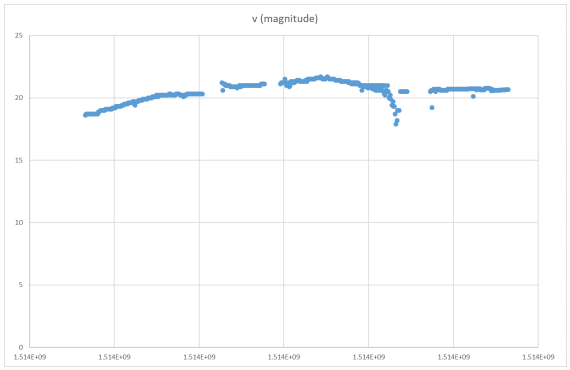
Figure. 43: Time – space diagram visualization tool. Example of grouping of vehicle traces



(a)

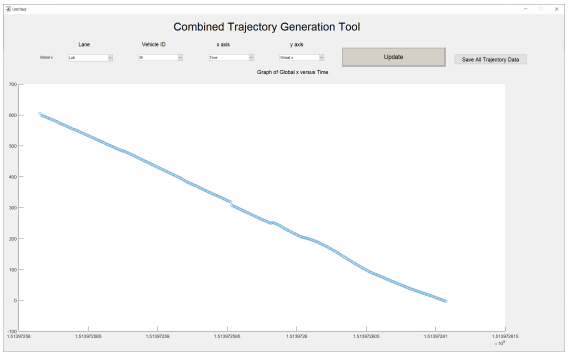


(b)

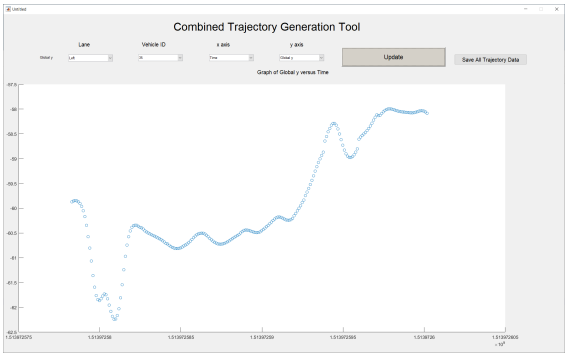


(c)

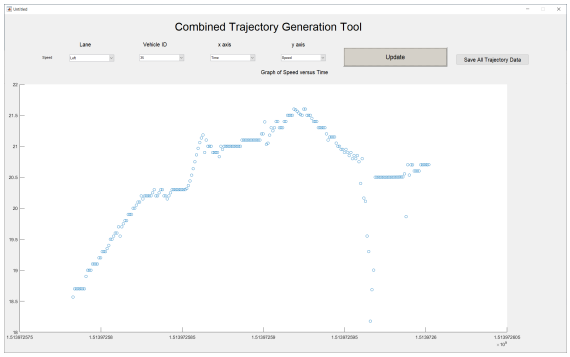
Figure. 44: Graphs of Raw group data



(a)



(b)



(c)

Figure. 45: Cleaned, combined, and Imputed actual vehicle trajectory information

8 Experiment using BSMs from Radar Data

The experiment with BSMs generated from the microscopic simulation provides an upper bound for the accuracy of the Kalman filter because the coordinates in BSMs from the microscopic simulation are always accurate. The BSMs generated from radar data include noises which may affect the estimation accuracy. The preparation of BSMs generated from radar data is introduced in section 7. A database including 100% of BSMs is created first, then four other databases including 20%, 40%, 60%, and 80% of BSMs are generated. Table 9 lists the sizes of generated databases. The estimated values of four parameters are shown in table 10.

Table 9: File sizes of BSM databases from radar data

Penetration rate	File size	Row number
20%	1.2 MB	15367
40%	2.25 MB	28990
60%	3.67 MB	47074
80%	4.81 MB	61615
100%	6.19 MB	79367

Figures 46 and 47 show the BSMs generated from radar data with different penetration rates. Based on Figure 46, the vehicles are not evenly distributed among three lanes, the third lane has the most vehicles while the first lane has the fewest vehicles.

Table 10: Traffic flow parameters for experiment 3

Parameter	Value
Free-flow speed v_f	62.3 miles per hour
Capacity Q	1909 vehicles per hour per lane
Shockwave Speed w	9.35 miles per hour
Jam density K	234 vehicles per mile per lane

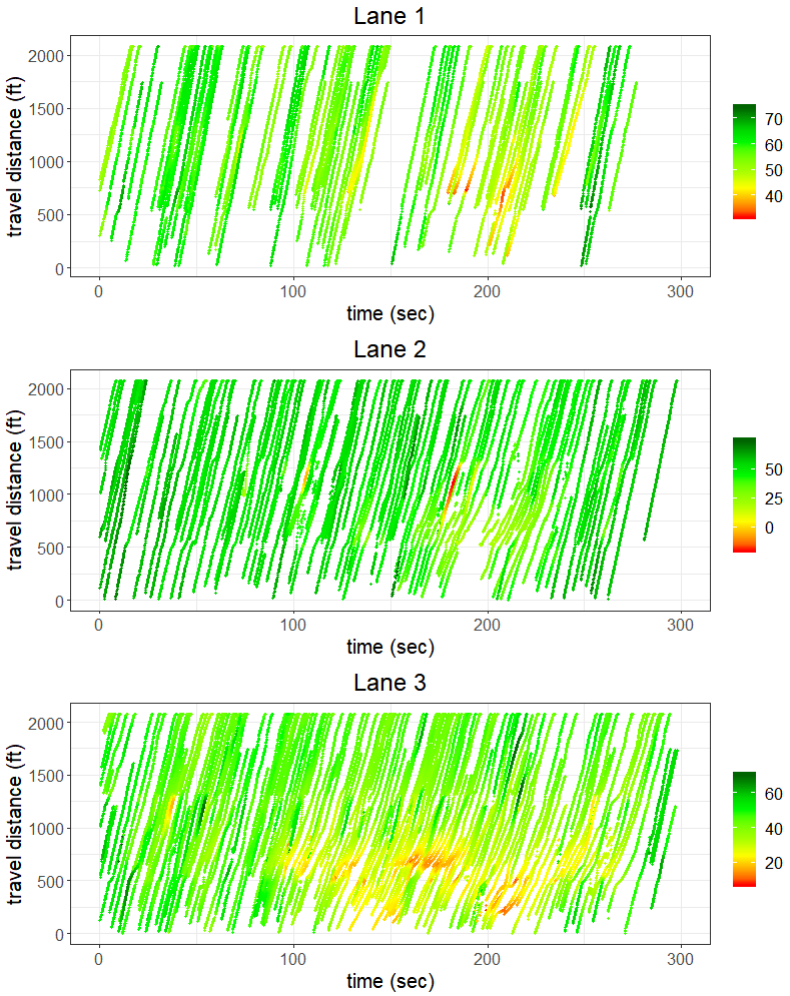
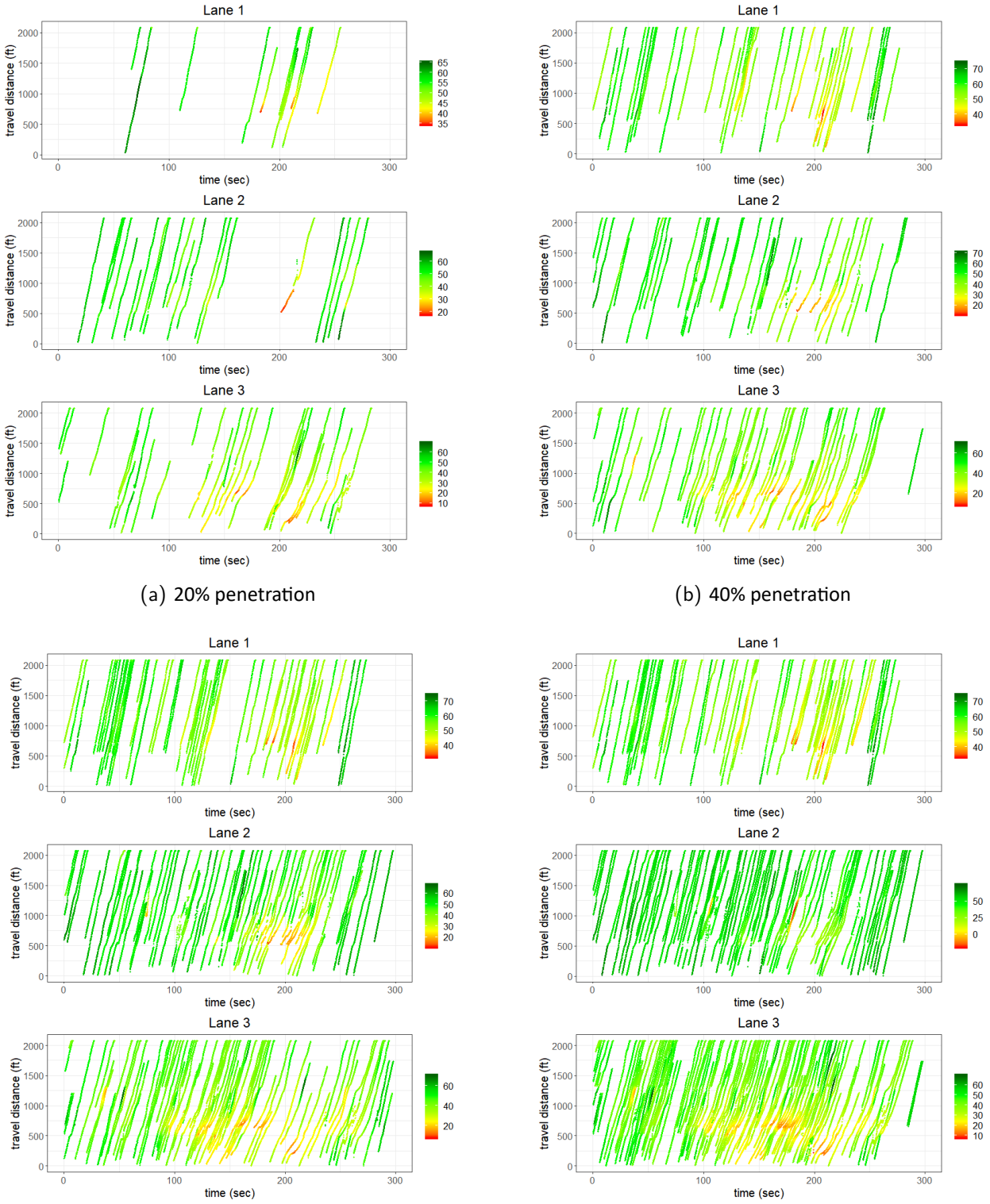


Figure. 46: 100% of BSMs from radar data



(a) 20% penetration

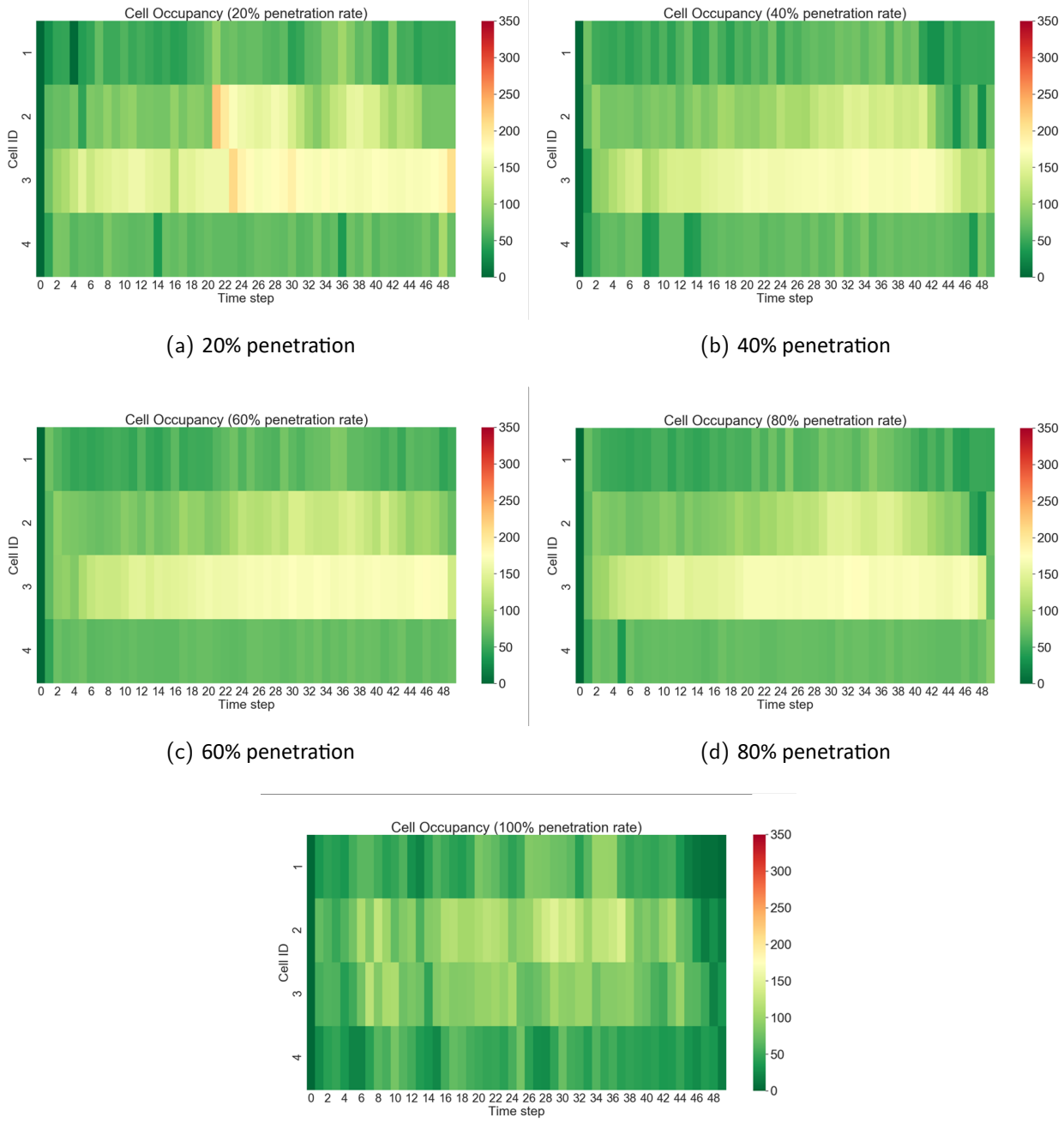
(b) 40% penetration

(c) 60% penetration

(d) 80% penetration

Figure. 47: BSMs from radar data

8.1 Traffic state estimation results



(a) 20% penetration

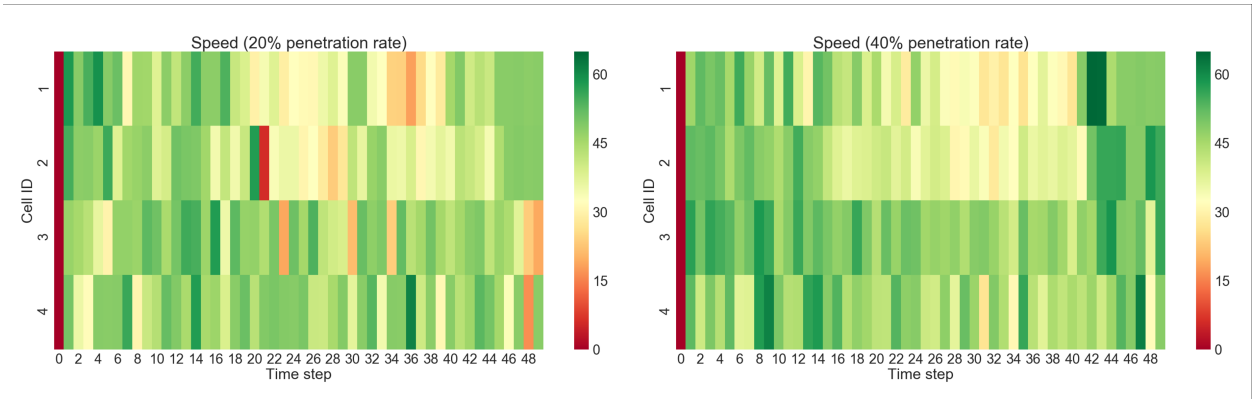
(b) 40% penetration

(c) 60% penetration

(d) 80% penetration

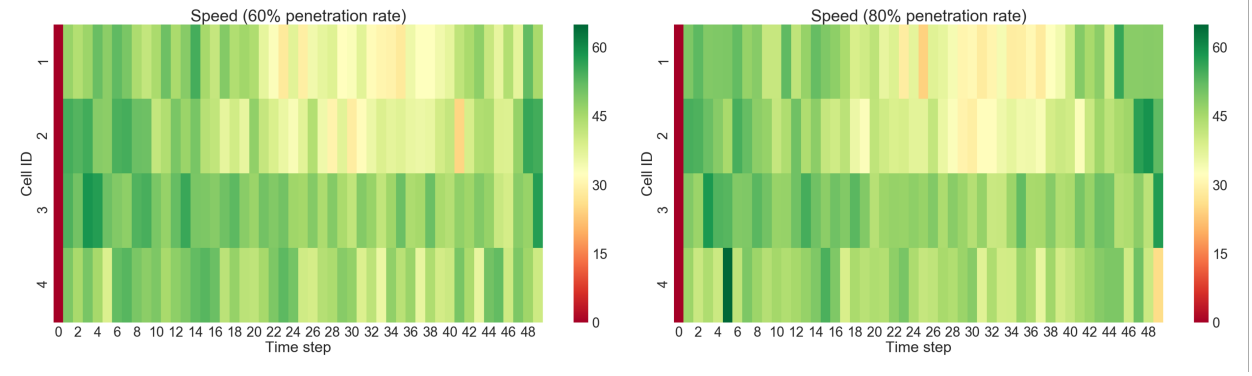
(e) 100% penetration

Figure. 48: Density estimation under different penetration rates with BSMs from radar data



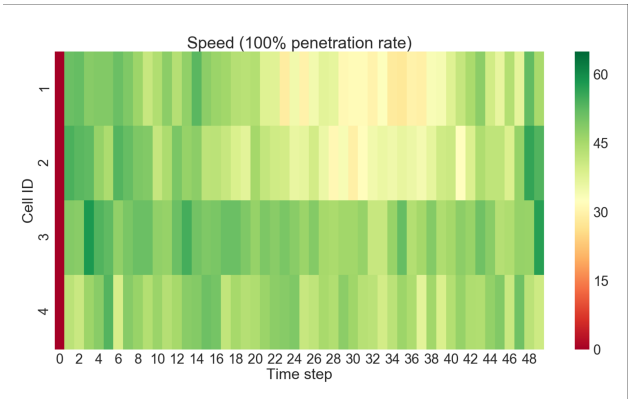
(a) 20% penetration

(b) 40% penetration



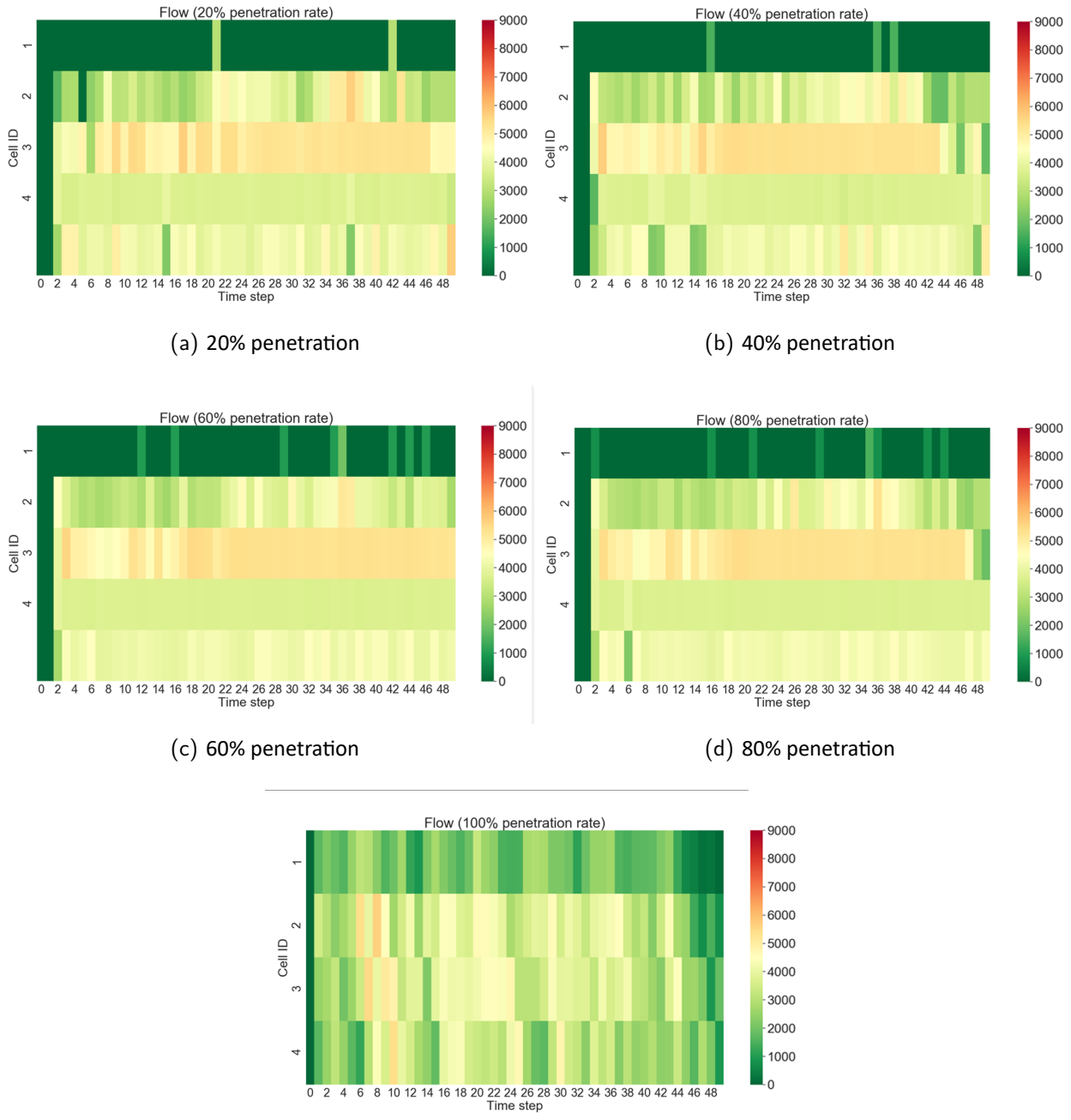
(c) 60% penetration

(d) 80% penetration



(e) 100% penetration

Figure. 49: Speed estimation under different penetration rates with BSMs from radar data



(a) 20% penetration

(b) 40% penetration

(c) 60% penetration

(d) 80% penetration

(e) 100% penetration

Figure. 50: Flow estimation under different penetration rates with BSMs from radar data

Figures 48, 49, and 50 show the traffic estimation results compared with the actual traffic states (figures with a 100% penetration). The difference between the estimated speeds and the actual speeds are small, but there are obvious differences between the estimated densities and esti-

mated flows, and actual densities and flows. In the heatmap of estimated densities, mild congestion forms at time step 5 on cell 3. Then the congestion propagates to cell 2 at time step 21. The congestion on cell 3 does not disappear until the end of the 5-min period. Compared with the heatmap of actual densities, the densities on cells 2, 3, and 4 are overestimated. In the heatmap of estimated speeds, the general speed variation is similar to the heatmap for actual speeds. The speed on cell 1 first drops at time step 22 as the cell color changes from light green to light yellow, followed by the speed drop on cell 2 at time step 24. In the heatmap of estimated flows, the first row is the input flow to cell 1. The second, third, fourth, and fifth rows are the flows going out of cells 1, 2, 3, and 4 respectively. The flows going out of four cells are overestimated as the color for most cells is light green in the heatmap of estimated flow. For cell 2 and cell 3, the estimated flows are almost constant and the flow going out of cell 2 stays at its capacity for a long time. The actual flows for these two cells have more variance as the cell color switches between green and yellow.

(a) Overall error

(b) Occupancy error

(c) Speed error

(d) Flow error

Figure. 51: Estimation errors under different penetration rates with BSMs from radar data

Figure 51 shows the error in the traffic state estimation with radar data. The error for speed

estimation is the smallest, with a maximum error of 13% when using a penetration rate of 20%. For density and flow estimations, the errors are about 51% and 52% respectively.

As we have seen in the estimation results with the BSMs from microsimulation models, the Kalman filter has relative low estimation accuracy under uncongested traffic conditions. As the BSMs from radar data corresponds to a 5-min uncongested time interval, the estimation accuracy here for the density and the flow is not high either. Based on the structure of the trapezoidal fundamental diagram, when the traffic is uncongested, given the speed, there is not a value of density uniquely corresponding to this value of speed. When the traffic is congested, there is a density that corresponds to a given value of the speed. Another reason that accounts for the low accuracy for the density and the flow is the small values of the actual density and speed. For example, if the actual density is 2 vehicles, and the estimated density is 4 vehicles, the error is as large as 100% even though the difference between the actual and the estimated density is 2 vehicles. If the actual density is 22 vehicles, and the estimated density is 20 vehicles, then the error is as small as 9.1%.

Besides, The error in speed estimation decreases with the increase of penetration rate, but the decreasing trend for density and flow estimations is not obvious when the penetration rate increases. The first cell has the largest error in both density and flow estimation. For the speed error, the difference among cells is not obvious.

The link travel times of vehicles are calculated using the estimated speed. The link travel time is calculated as the summation of the travel time of each cell, as shown in equation (32). The travel time estimations under all penetration rates show high accuracy. The link travel time is about 28 seconds at first and then increases to 40 seconds before it drops to 28 seconds in the end.

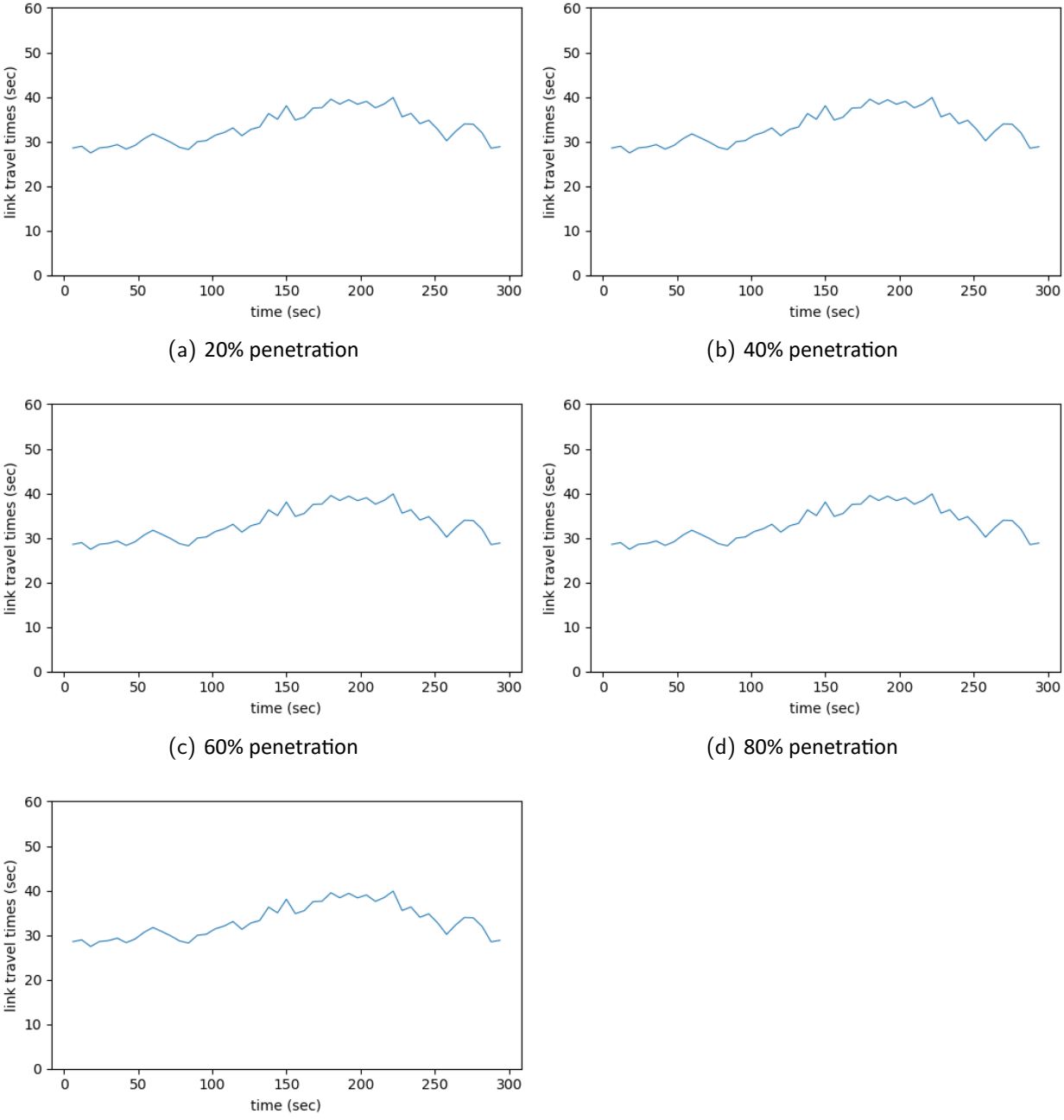
Besides, the queue length of the road is estimated using the estimated density of $\hat{n}_i$ and speed $\hat{v}_i$. Figure 53 shows that the estimated queue length is zero. The queue length is overestimated when 20% and 40% of BSMs are used.

8.2 Discussion

The estimation results show that the accuracy of the Kalman filter is good when applying to the BSMs generated by microscopic simulation models. Under the uncongested scenario, errors for the density, the speed, and the flow estimations are about 23%, 3%, and 23% respectively. Under the congested scenario, errors for the density, the speed, and the flow estimations are about 15%, 3%, and 12% respectively. An increase in the penetration rate increases the estimation accuracy of the Kalman filter.

The accuracy of the Kalman filter is much lower when applied to BSMs from radar data as

there are noises in the BSMs. The density and the flow estimations have an error of about 51%. Because the values for actual densities and flows are small, so the estimation error for these two variables is large. For speed estimation, the error is as small as 8%. Besides, the queue length and the link travel times are also estimated with the estimated traffic states. Both the link travel time estimation and the queue length estimation show high accuracy.



(a) 20% penetration

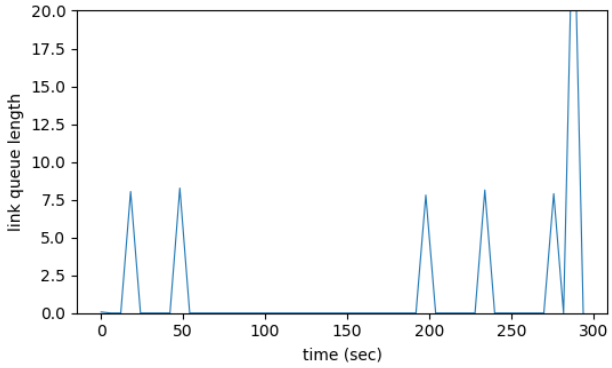
(b) 40% penetration

(c) 60% penetration

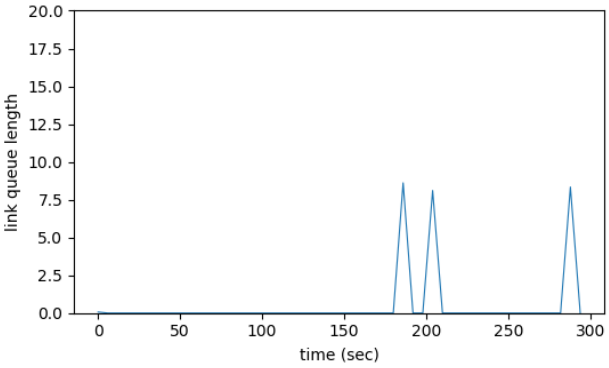
(d) 80% penetration

(e) 100% penetration

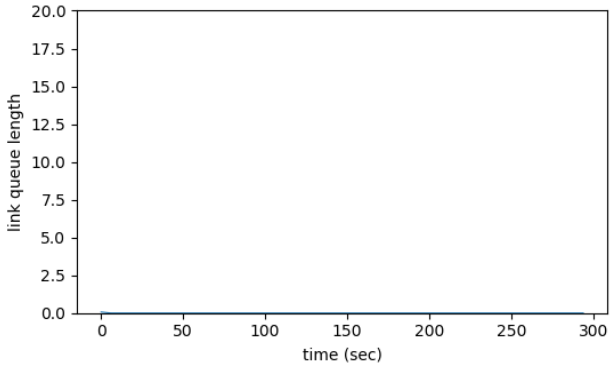
Figure. 52: Link travel time estimation under different penetration rates



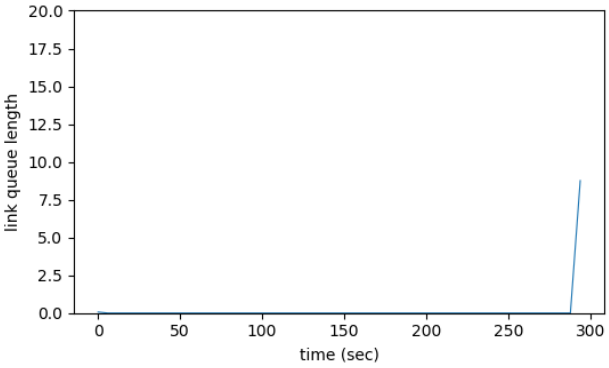
(a) 20% penetration



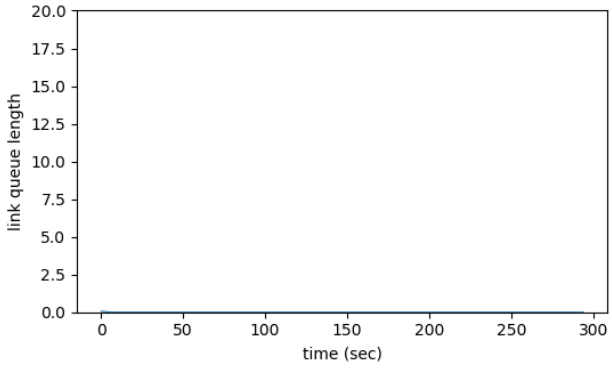
(b) 40% penetration



(c) 60% penetration



(d) 80% penetration



(e) 100% penetration

Figure. 53: Link queue length estimation under different penetration rates

9 Tests with Detectors Measuring Input Flows

9.1 Input flow measurement

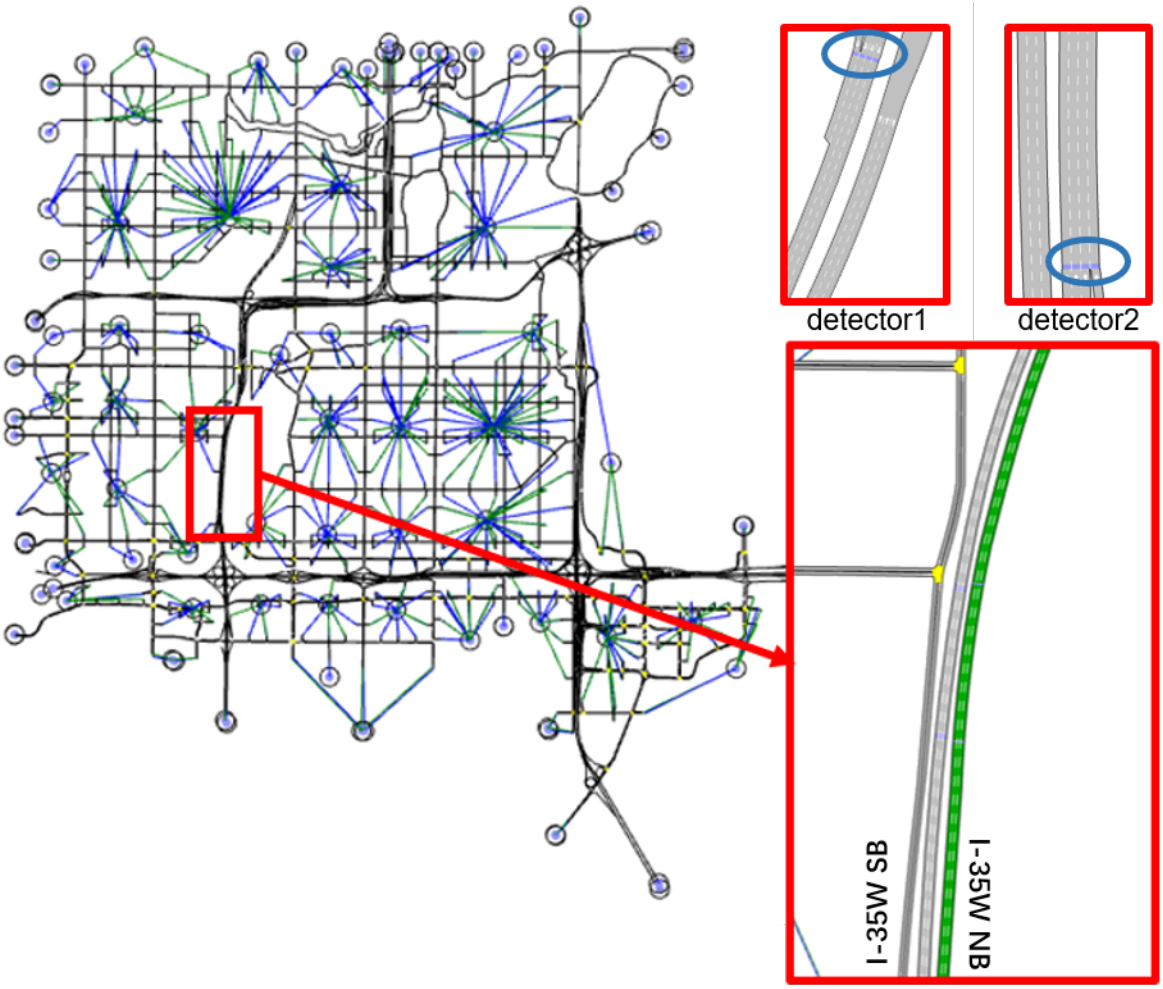


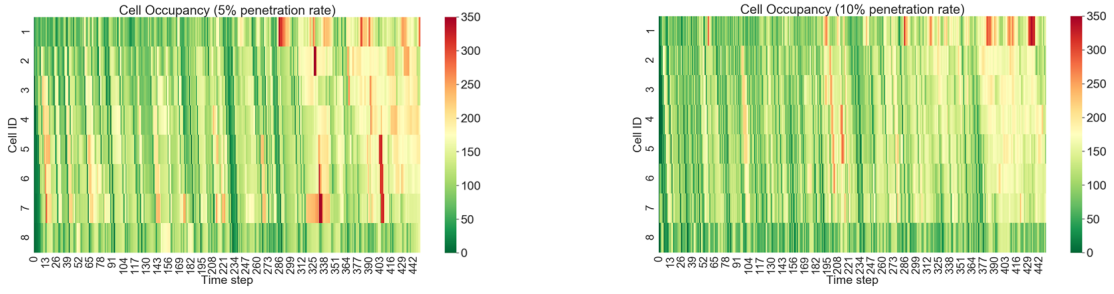
Figure. 54: Target road sections in AIMSUN

For experiment using AIMSUN, detectors at one end of freeways are used to collect the input flow. Figure 54 shows the location of two target road sections, which are northbound and southbound of freeway I-35W. In AIMSUN, one road is congested but the other is not during the morning peak hours (6:45am - 7:30am), which provide the data for both congested scenario and uncongested scenario. Two sub-figures at the upper right of Figure 54 shows the location of two detectors on target road sections to measure the input flow. Detector 1 is on southbound of freeway I-35W and detector 2 is on northbound of freeway I-35W. Because the time step for the Kalman filter is set to be 6 seconds, the number of vehicles entering the link is measured every 6 seconds.

For BSMs generated from radar data, to get the measurement of the input flow, we need to find the detector located at the beginning of the target road section. However, there is no detector installed on freeway that is at the edge of zone A (shown in Figure **??**). Then the original radar data including 100% vehicles is processed to get the input flow to the road section.

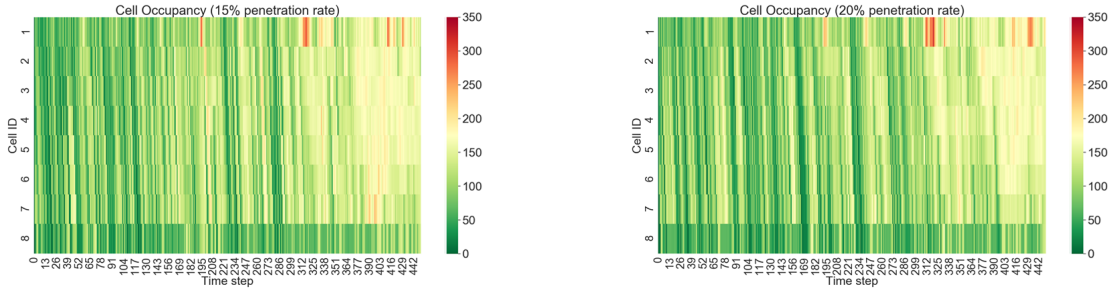
9.2 BSMs from AIMSUN under uncongested scenario

9.2.1 Density estimation



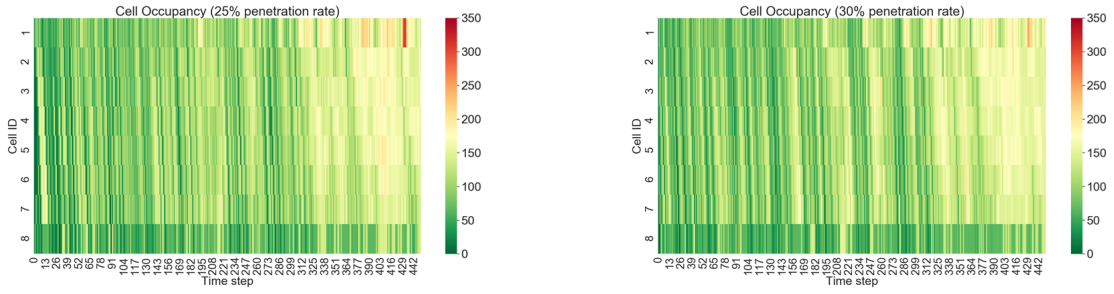
(a) 5% penetration

(b) 10% penetration



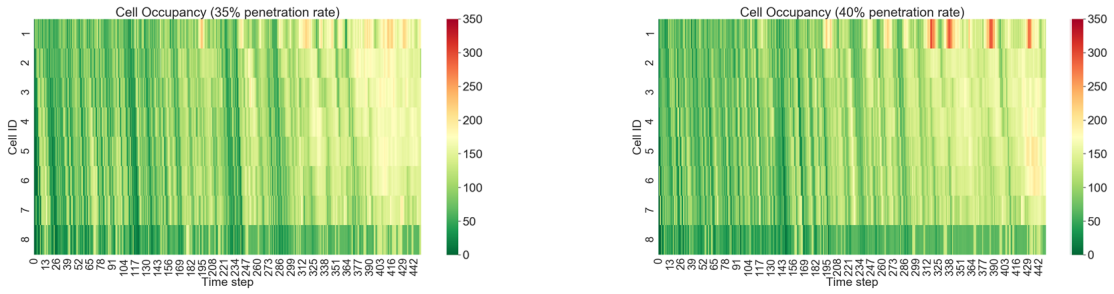
(c) 15% penetration

(d) 20% penetration



(e) 25% penetration

(f) 30% penetration



(g) 35% penetration

(h) 40% penetration

Figure. 55: Density estimation with BSMs from AIMSUN (uncongested condition)

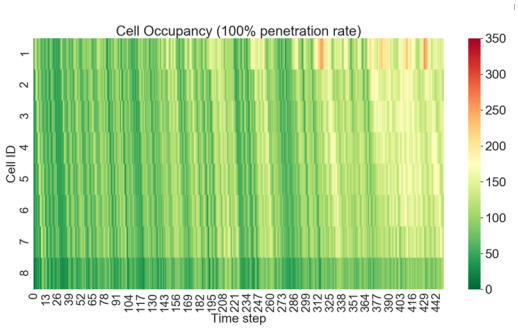


Figure. 56: Actual density in uncongested condition (100% vehicles)

9.2.2 Speed estimation

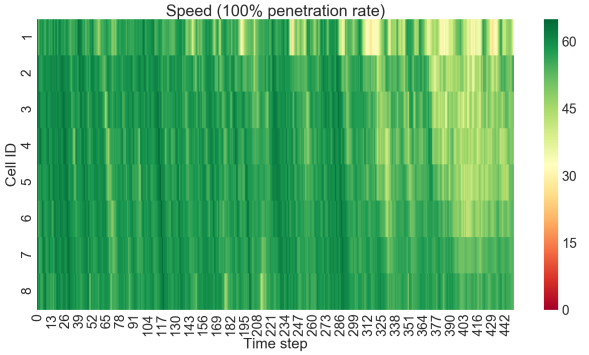
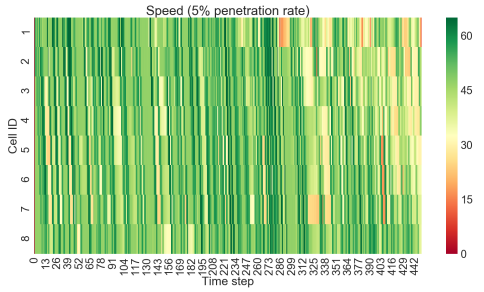
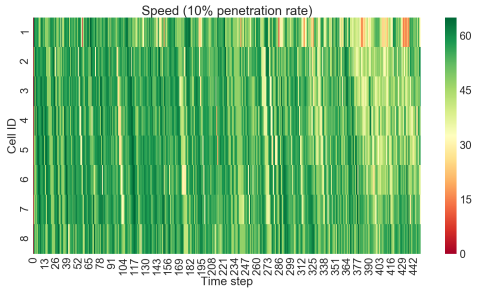


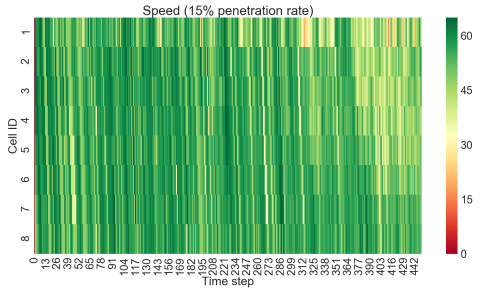
Figure. 57: Actual speed in uncongested condition (100% vehicles)



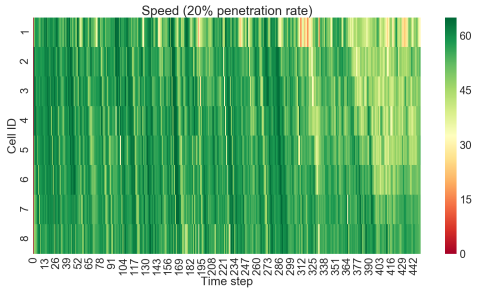
(a) 5% penetration



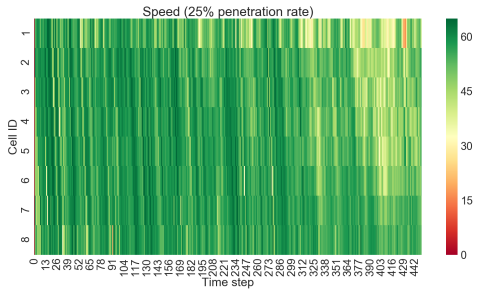
(b) 10% penetration



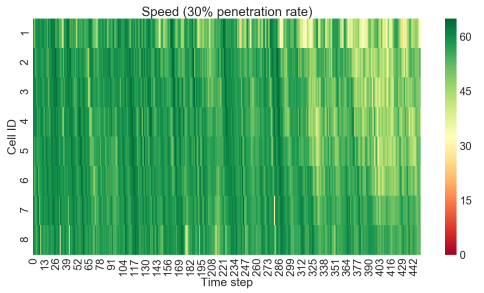
(c) 15% penetration



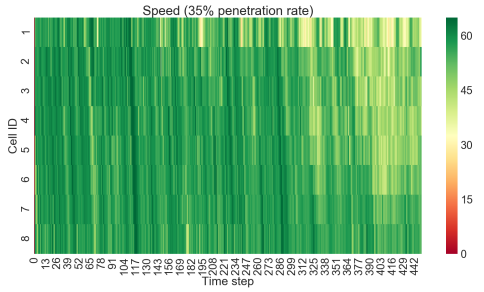
(d) 20% penetration



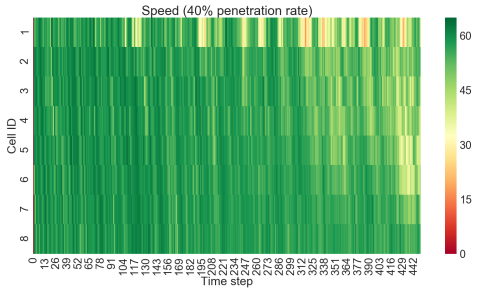
(e) 25% penetration



(f) 30% penetration



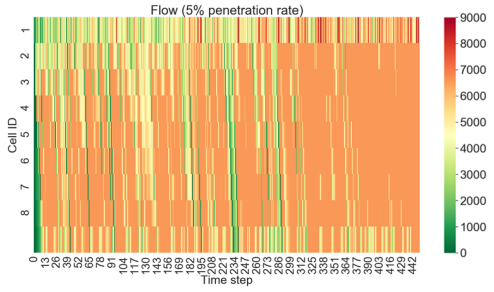
(g) 35% penetration



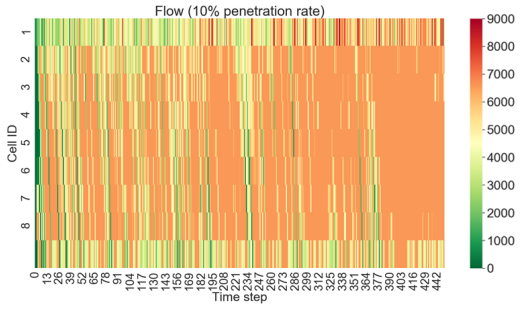
(h) 40% penetration

Figure. 58: Speed estimation with BSMs from AIMSUN (uncongested condition)

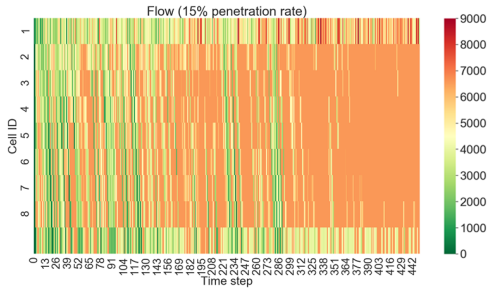
9.2.3 Flow estimation



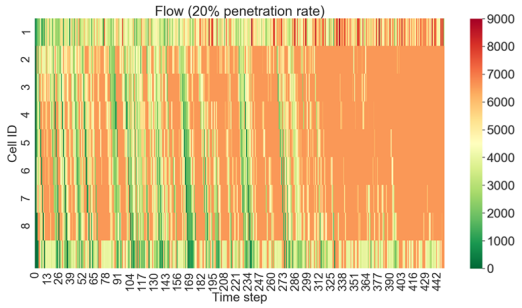
(a) 5% penetration



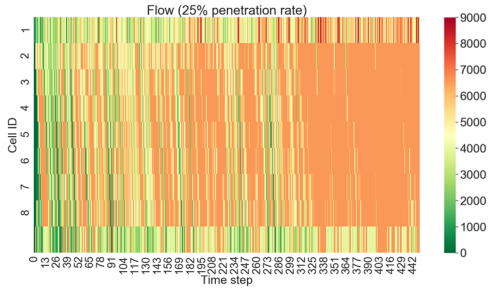
(b) 10% penetration



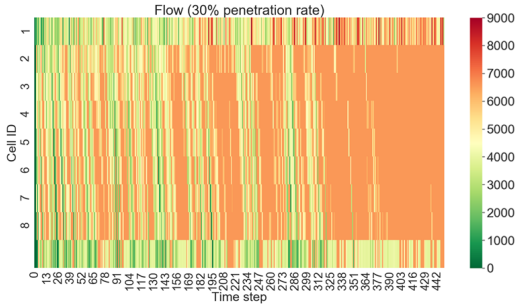
(c) 15% penetration



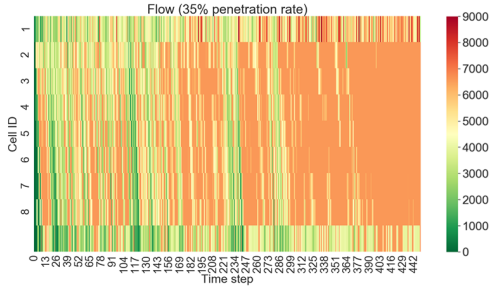
(d) 20% penetration



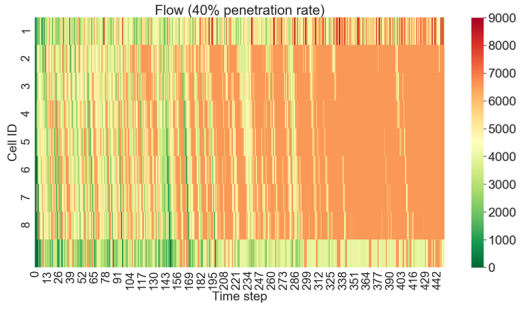
(e) 25% penetration



(f) 30% penetration



(g) 35% penetration



(h) 40% penetration

Figure. 59: Flow estimation with BSMs from AIMSUN (uncongested condition)

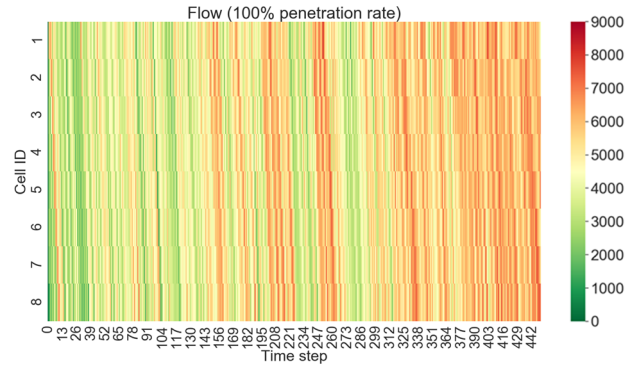


Figure. 60: Actual flow in uncongested condition (100% vehicles)

9.2.4 Estimation error

(a) Overall error

(b) Density error

(c) Speed error

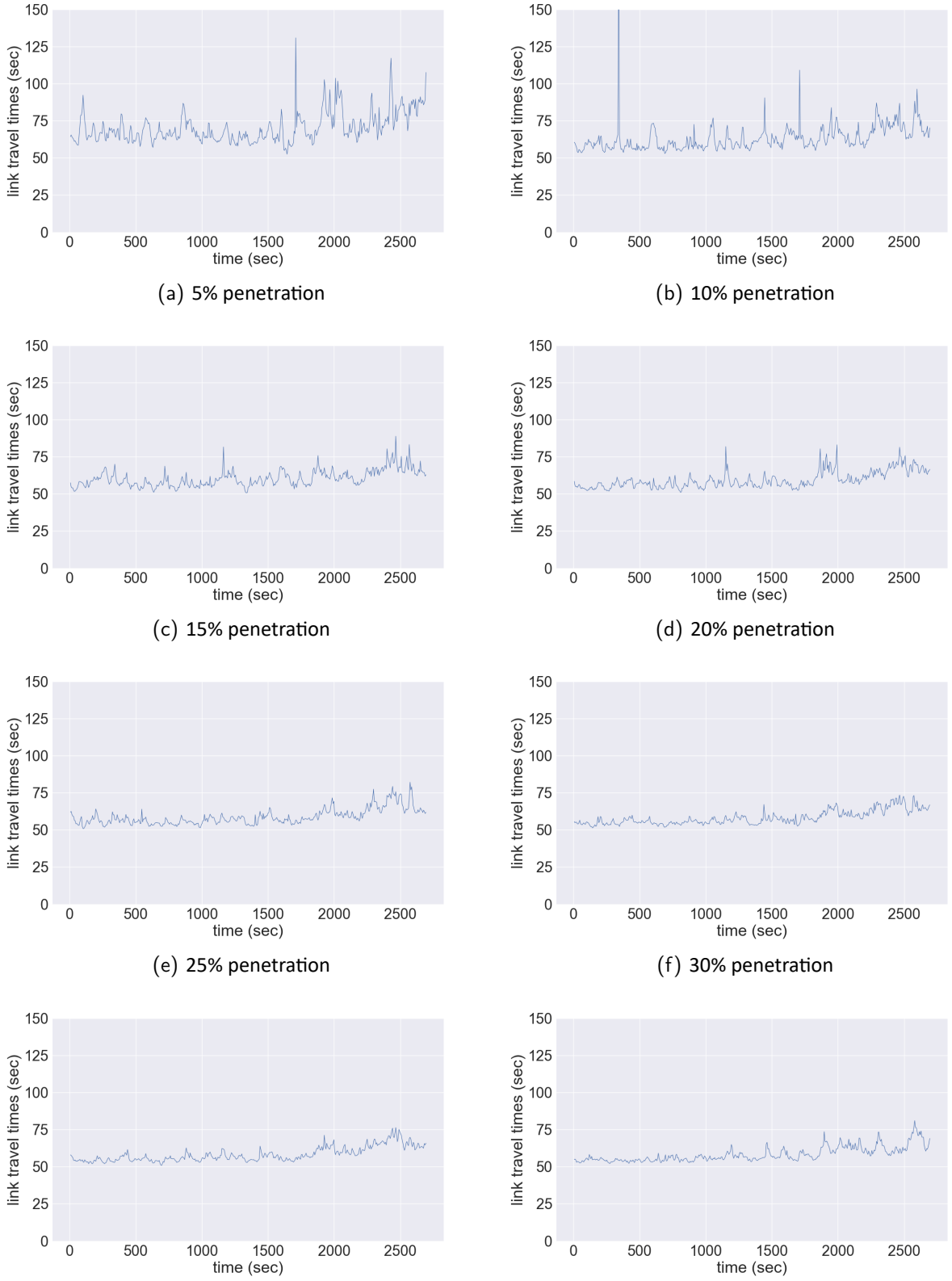
(d) Flow error

Figure. 61: Estimation errors with BSMs from AIMSUN (uncongested condition)

Figure 61 shows the error for the experiment under uncongested conditions using detector data for the first cell. As in Task 3, the speed error is the lowest as it is obtained through direct measure-

ment, while density error is the highest. In Task 3 there is a sharp decrease in error between cells 1 and 2 that is not seen in the density and flow error charts in Task 4. It seems that the addition of detector observations in Task 4 dramatically reduce the error in cell 1 in both the density and flow estimates. The reduction in error in the first cell is reflected in the reduction in the mean error of flow and density, especially in the first few cells. Between Task 3 and Task 4, the mean density error drops from 48% to 32% for 5% market penetration, 40% to 32% for 10% market penetration, and 33% to 29% at 15% market penetration, and by about 3% for market penetrations of 20% to 30%. Likewise, the mean flow error drops by around 5% for 5% market penetration, and between 1% and 4% for 10% to 30% market penetration. The error for speed seems to be largely unchanged between Task 3 and Task 4.

9.2.5 Travel time estimation



(a) 5% penetration

(b) 10% penetration

(c) 15% penetration

(d) 20% penetration

(e) 25% penetration

(f) 30% penetration

(g) 35% penetration

(h) 40% penetration

Figure. 62: Travel time estimation with BSMs from AIMSUN (uncongested condition)

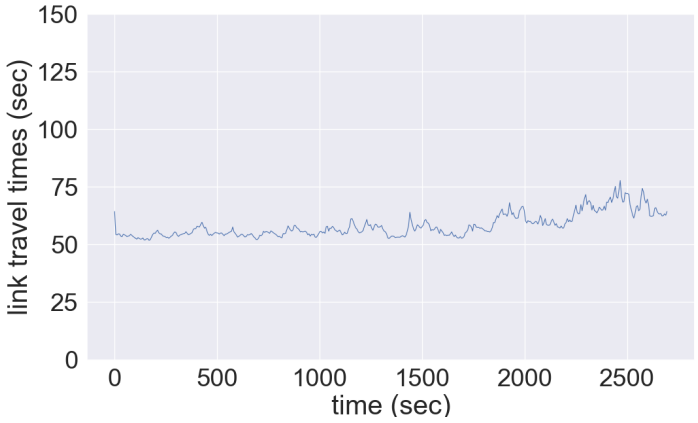


Figure. 63: Actual link travel time in uncongested condition (100% vehicles)

The link travel times are estimated using the estimated speed. It is calculated by summing up travel times in all cells. The estimation of link travel times show high accuracy. Under all penetration rates, the travel time is initially about 52 seconds, and then increases to about 70 seconds. There is no large difference between the travel time estimations in Task 3 and Task 4 so that adding detector data to the first cell does not help increasing the accuracy of travel time estimation.

9.2.6 Queue length estimation

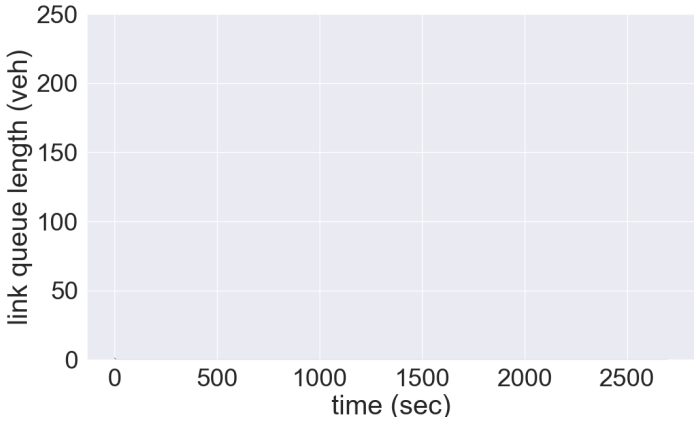
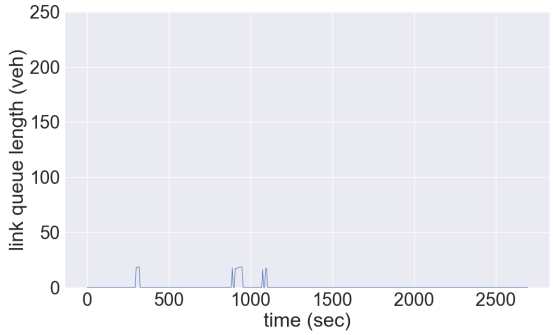


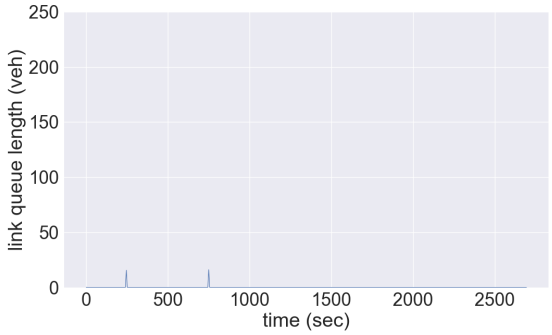
Figure. 64: Actual link queue length in uncongested condition (100% vehicles)

Queue length is also estimated, which is shown in 65. According to Figure 64, in uncongested scenario, there is no queue formed on the freeway. In Figure 65, when the market penetration is

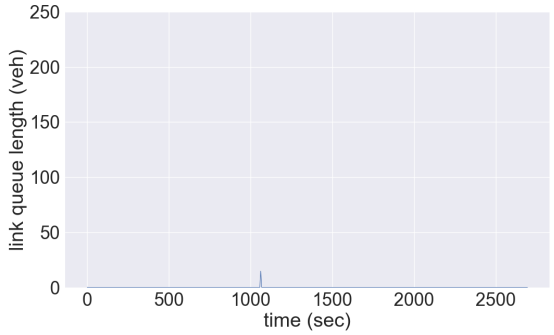
small, some short queue are formed on freeway according to the estimation result. Comparing the estimation results in Task3 and Task4, there is no big difference.



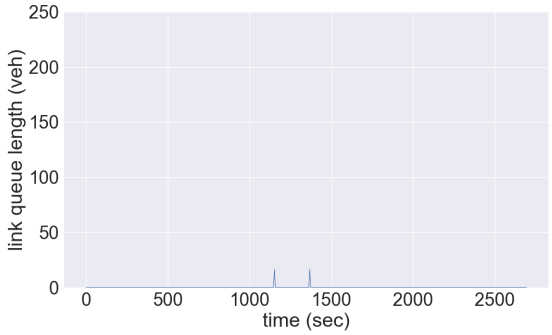
(a) 5% penetration



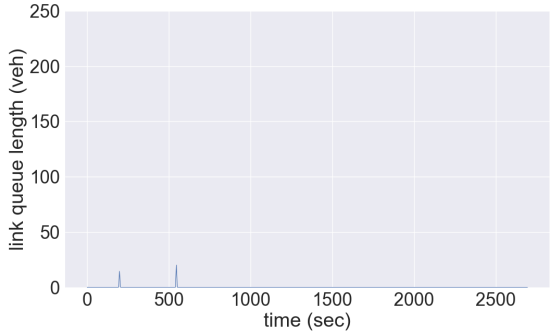
(b) 10% penetration



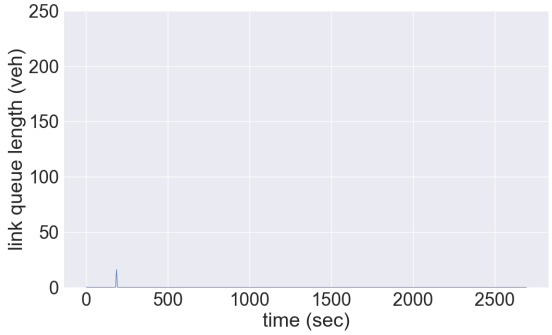
(c) 15% penetration



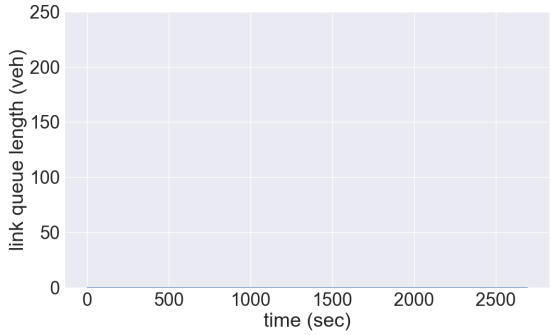
(d) 20% penetration



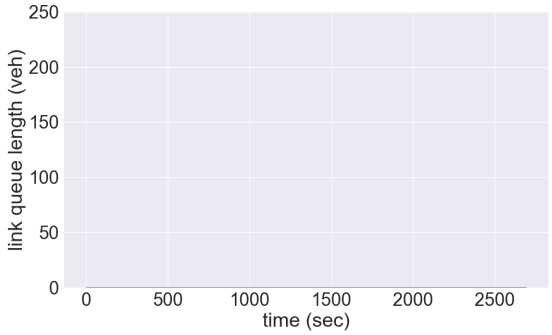
(e) 25% penetration



(f) 30% penetration



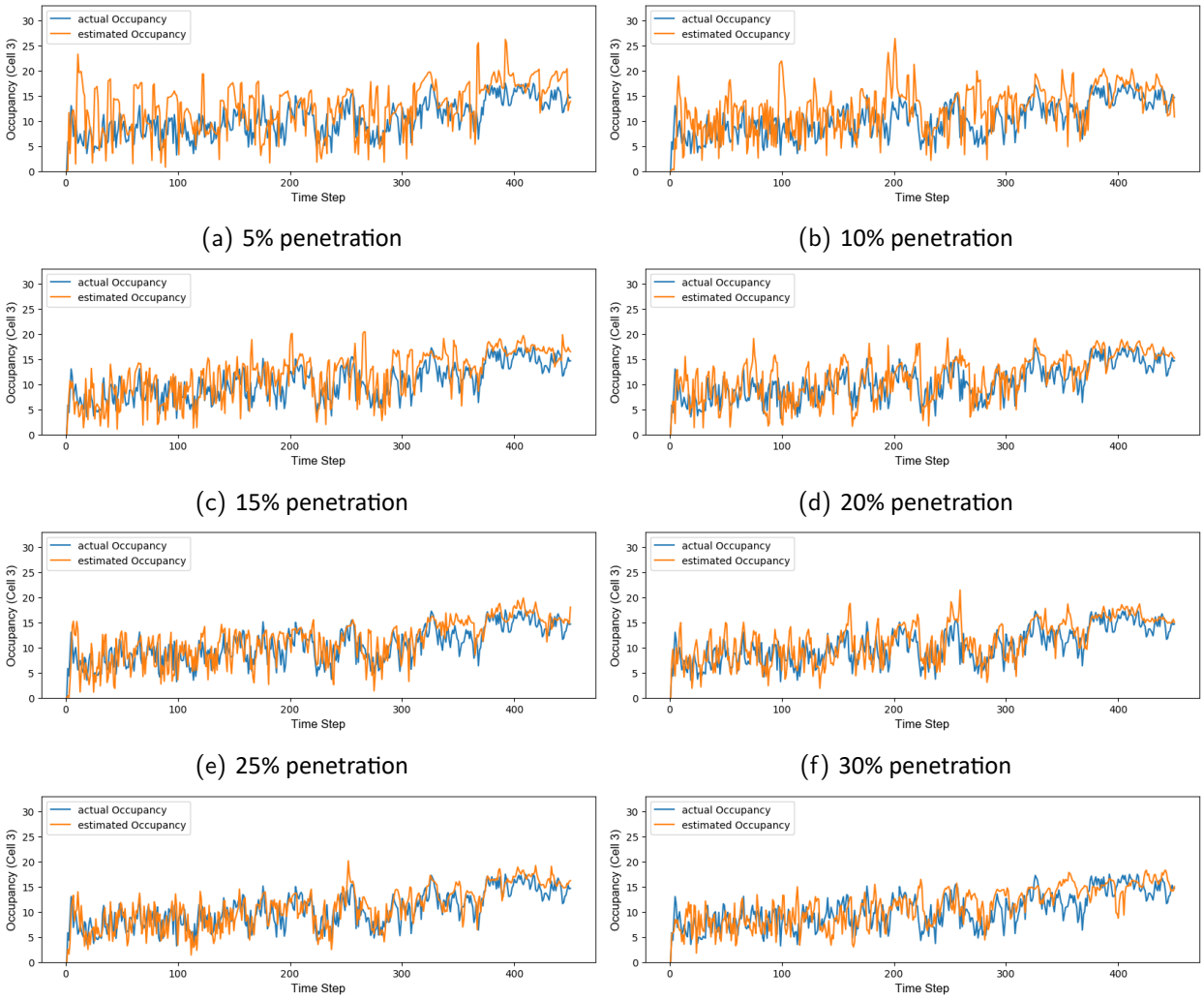
(g) 35% penetration



(h) 40% penetration

Figure. 65: Queue length estimation with BSMs from AIMSUN (uncongested condition)

9.2.7 The effect of increasing penetration rates



(a) 5% penetration

(b) 10% penetration

(c) 15% penetration

(d) 20% penetration

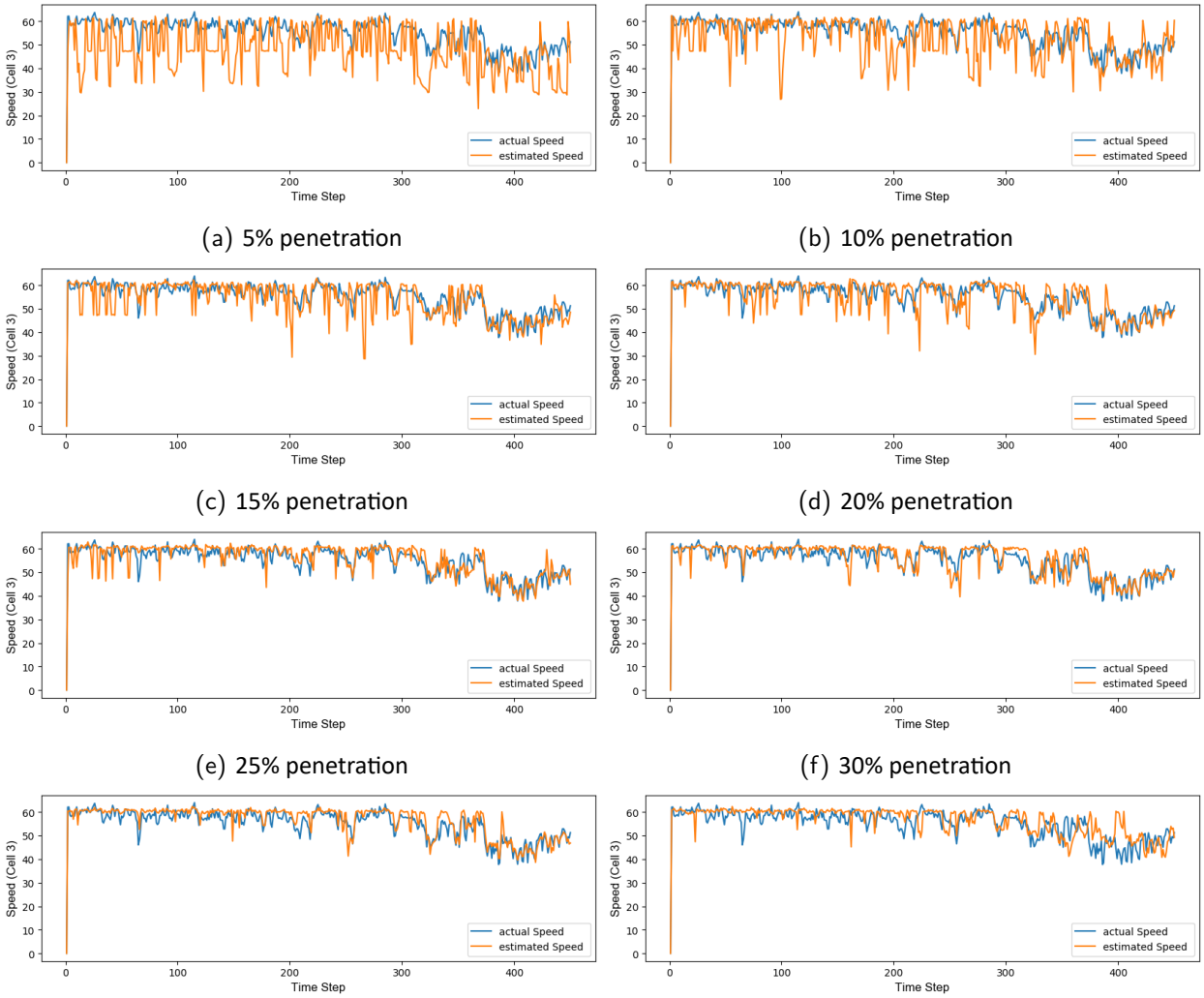
(e) 25% penetration

(f) 30% penetration

(g) 35% penetration

(h) 40% penetration

Figure. 66: Effect of penetration rates to density estimation for cell 3 (congested condition)



(a) 5% penetration

(b) 10% penetration

(c) 15% penetration

(d) 20% penetration

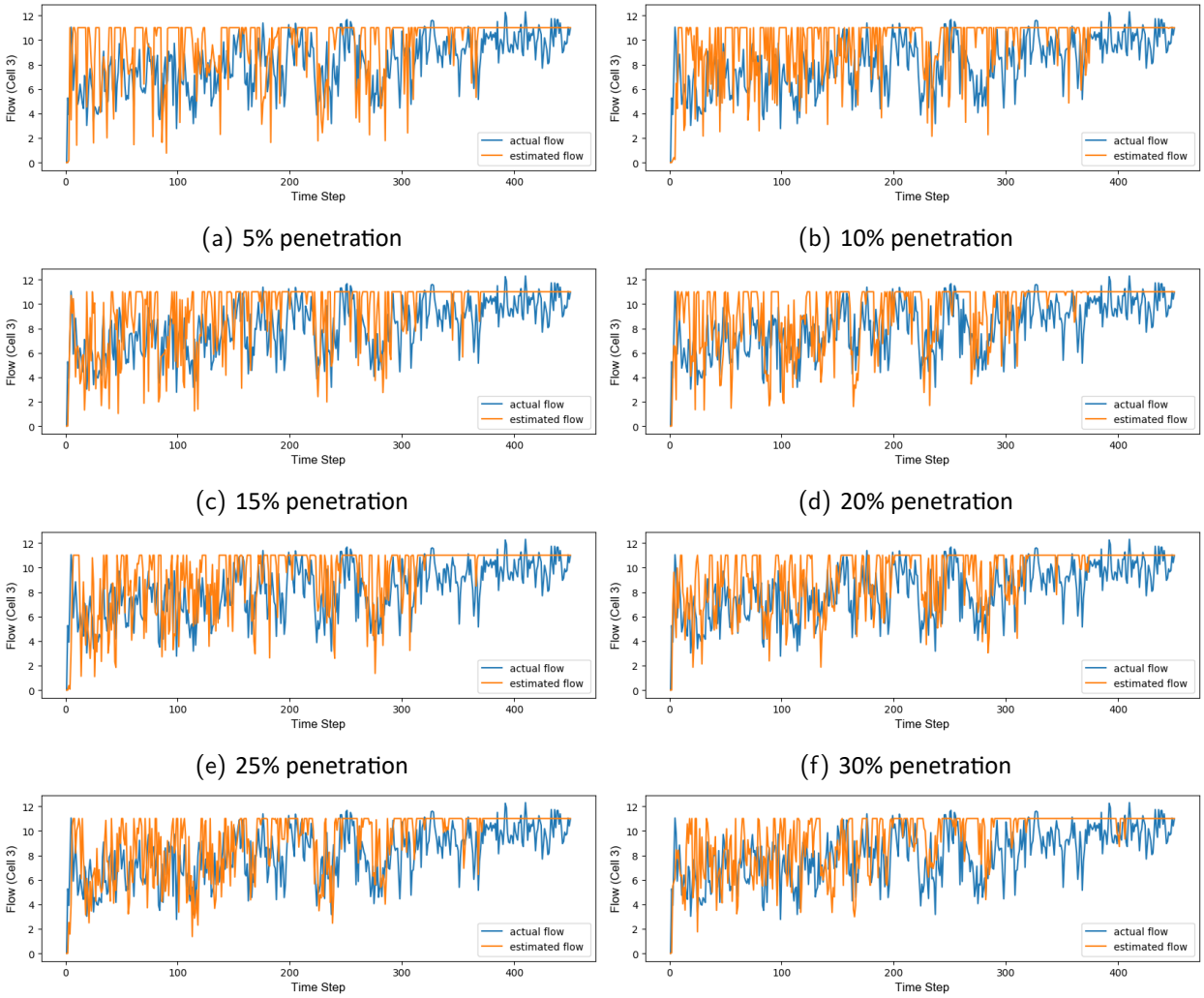
(e) 25% penetration

(f) 30% penetration

(g) 35% penetration

(h) 40% penetration

Figure. 67: Effect of penetration rates to speed estimation for cell 3 (congested condition)



(a) 5% penetration

(b) 10% penetration

(c) 15% penetration

(d) 20% penetration

(e) 25% penetration

(f) 30% penetration

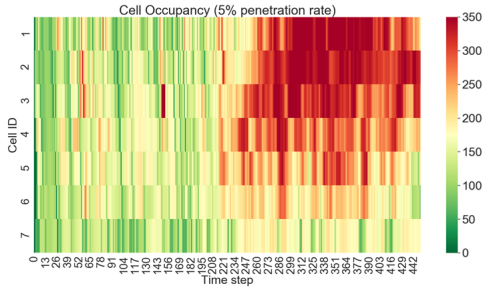
(g) 35% penetration

(h) 40% penetration

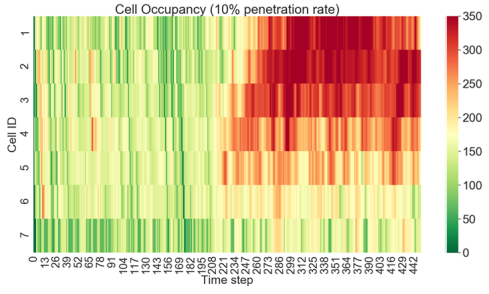
Figure. 68: Effect of penetration rates to flow estimation for cell 3 (congested condition)

9.3 BSMs from AIMSUN under congested scenario

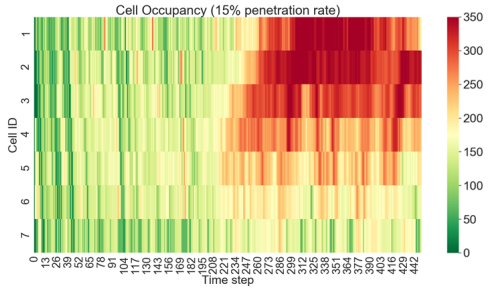
9.3.1 Density estimation



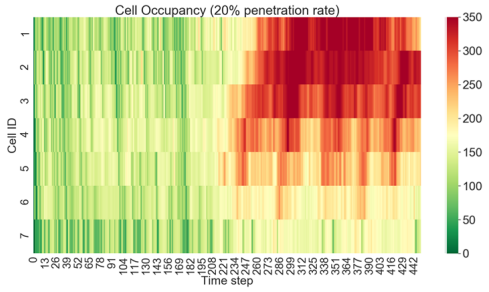
(a) 5% penetration



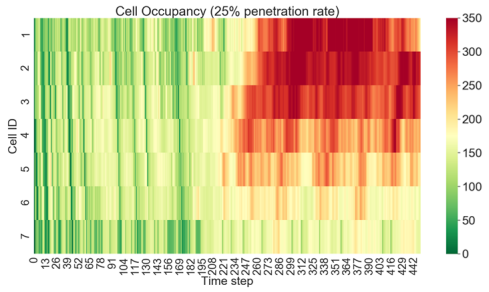
(b) 10% penetration



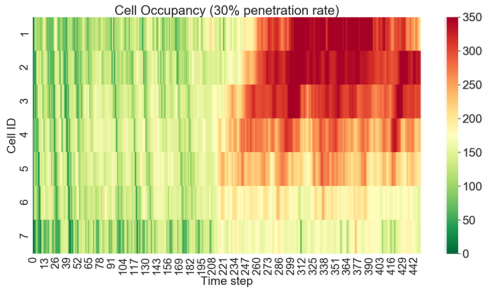
(c) 15% penetration



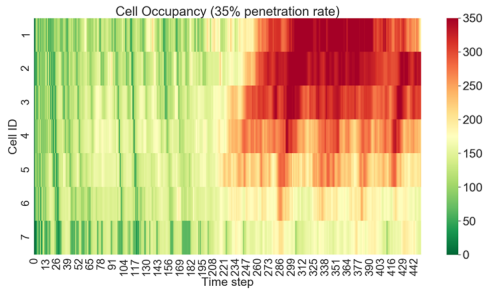
(d) 20% penetration



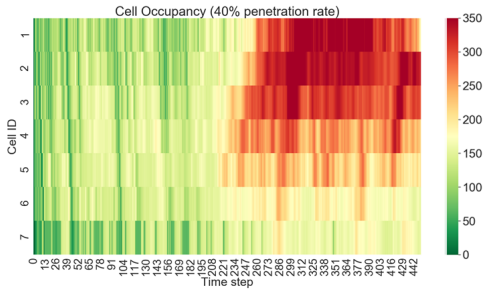
(e) 25% penetration



(f) 30% penetration



(g) 35% penetration



(h) 40% penetration

Figure. 69: Density estimation with BSMs from AIMSUN (congested condition)

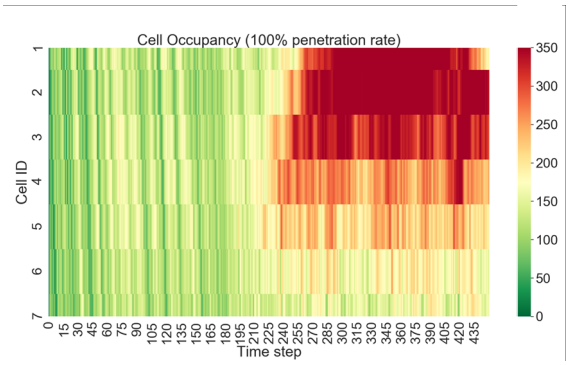


Figure. 70: Actual density in congested condition (100% vehicles)

9.3.2 Speed estimation

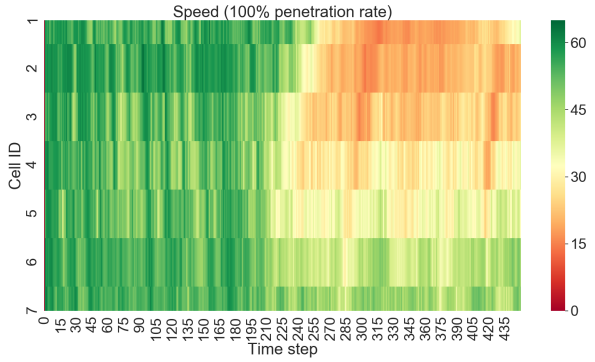
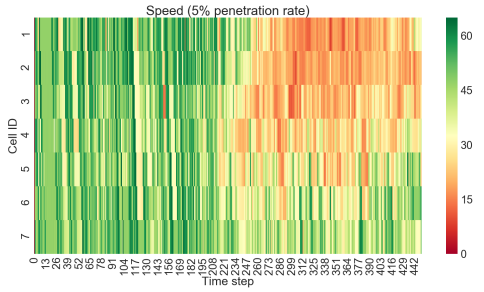
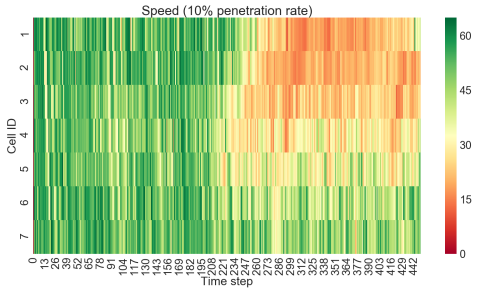


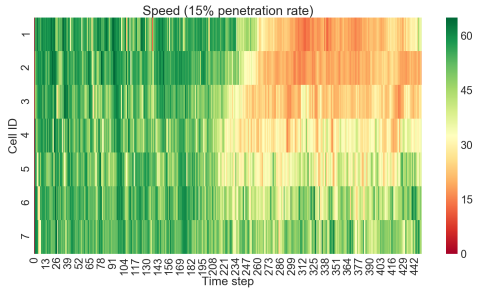
Figure. 71: Actual speed in congested condition (100% vehicles)



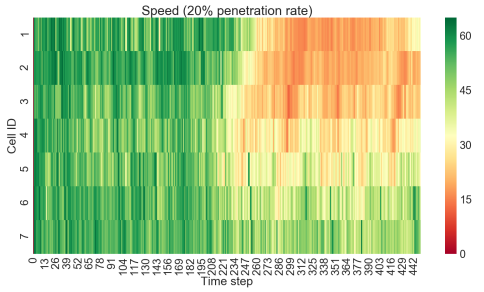
(a) 5% penetration



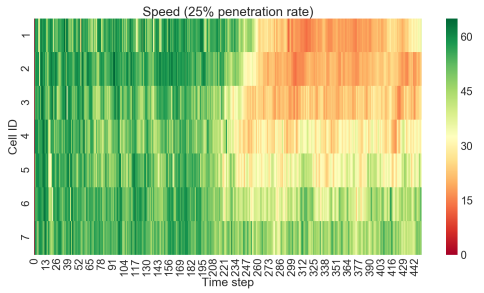
(b) 10% penetration



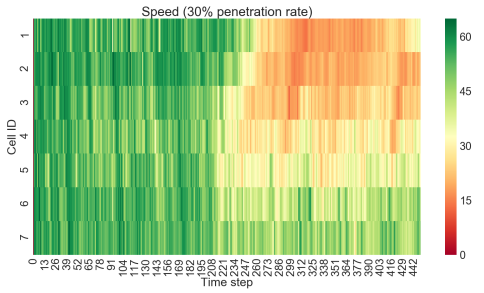
(c) 15% penetration



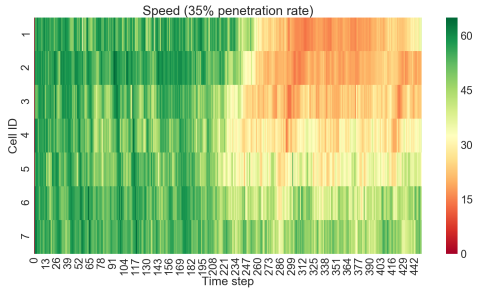
(d) 20% penetration



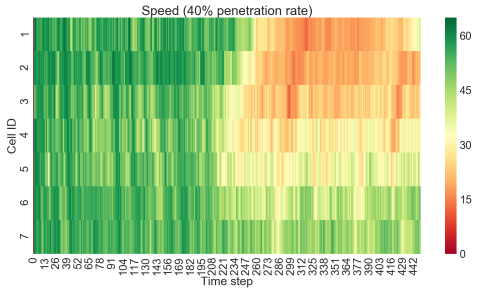
(e) 25% penetration



(f) 30% penetration



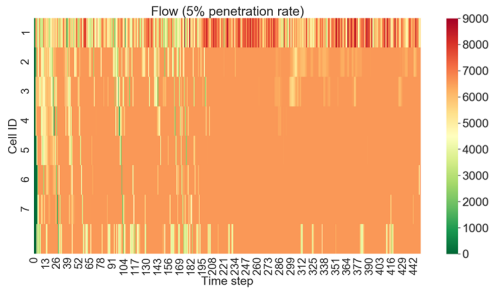
(g) 35% penetration



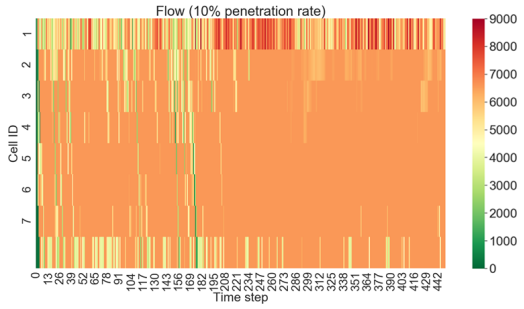
(h) 40% penetration

Figure. 72: Speed estimation with BSMs from AIMSUN (congested condition)

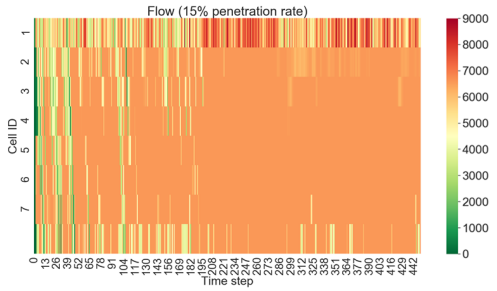
9.3.3 Flow estimation



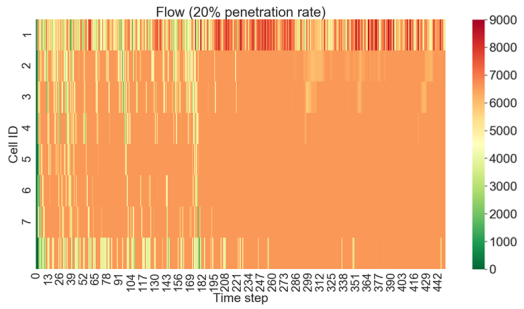
(a) 5% penetration



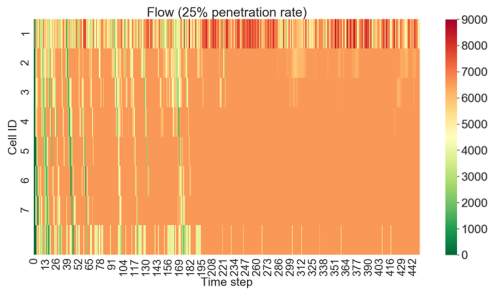
(b) 10% penetration



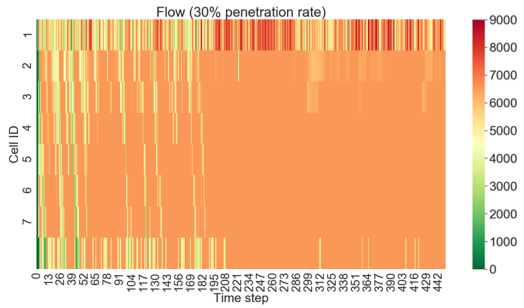
(c) 15% penetration



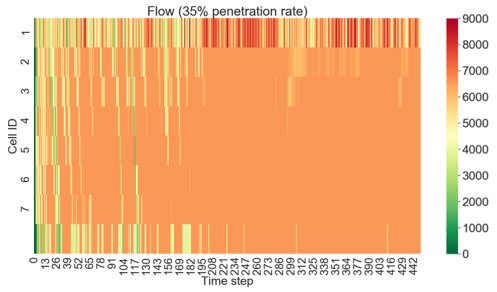
(d) 20% penetration



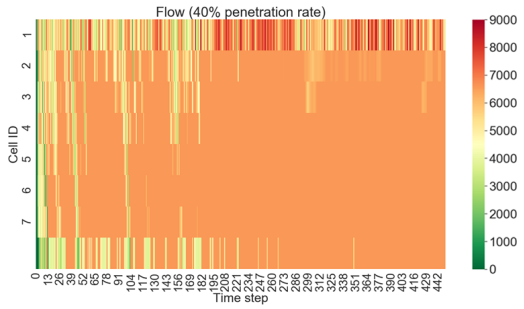
(e) 25% penetration



(f) 30% penetration



(g) 35% penetration



(h) 40% penetration

Figure. 73: Flow estimation with BSMs from AIMSUN (congested condition)

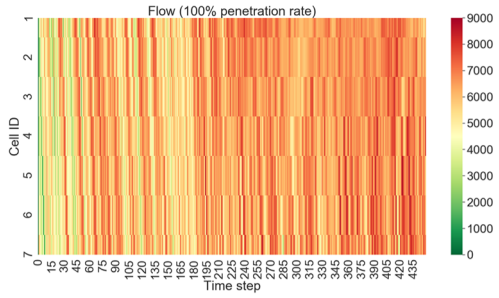


Figure. 74: Actual flow in congested condition (100% vehicles)

9.3.4 Estimation error

(a) Overall error

(b) Density error

(c) Speed error

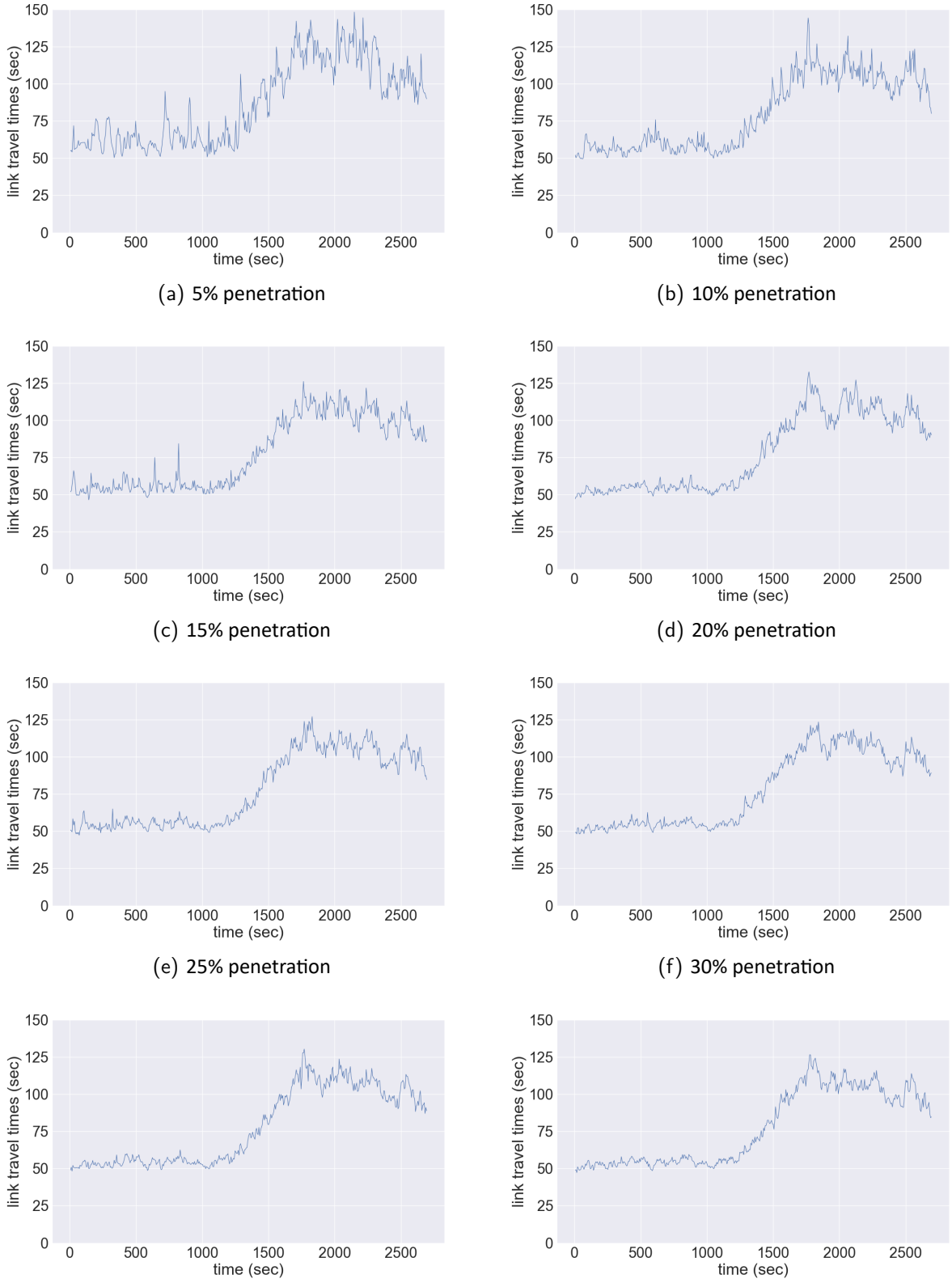
(d) Flow error

Figure. 75: Estimation errors with BSMs from AIMSUN (congested condition)

Figure 75 reveals that the trends found in the congested scenario mostly reflect those found in the uncongested scenario. The addition of the detector measurements decreases the error for

density and flow by significantly improving accuracy within the first cell while not significantly affecting the error for speed. However, the improvement to the flow error from Task 3 to Task 4 is much more pronounced in the congested scenario than in the uncongested scenario. In the uncongested scenario, the mean flow error from Task 3 to Task 4 decreased by 1 to 5 points depending on the market penetration, whereas in the uncongested scenario the mean flow drops from 30% to 15% at 5% market penetration and drops from 17% to 14% at 30% market penetration. The drop in mean flow error seems to be around 3 times larger in magnitude in the congested scenario compared to the uncongested scenario. The improvement in the density error in the congested scenario from Task 3 to Task 4 is less significant at lower market penetrations than it is in the uncongested scenario, as the error dropped by at most 5 percentage points compared to the maximum drop of 16 percentage points in the uncongested scenario, due to the comparatively low error in the Task 3 congested scenario compared to the Task 3 uncongested scenario.

9.3.5 Travel time estimation



(a) 5% penetration

(b) 10% penetration

(c) 15% penetration

(d) 20% penetration

(e) 25% penetration

(f) 30% penetration

(g) 35% penetration

(h) 40% penetration

Figure. 76: Travel time estimation with BSMs from AIMSUN (congested condition)

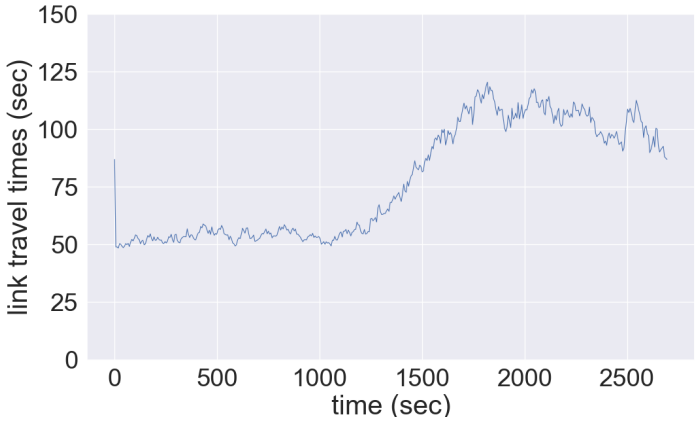


Figure. 77: Actual link travel time in congested condition (100% vehicles)

The link travel times are estimated, as shown in Figure 76. It starts from 50 seconds and then increases to about 120 seconds before it drops to 90 seconds. When the penetration rate is small, the travel time is overestimated which reflects the error in travel speed estimation is larger when the penetration rate is smaller.

9.3.6 Queue length estimation

Figure 79 shows the queue length estimation. According to actual values of queue length, queues are formed on freeway three times at about 1650, 2250, and 2600 seconds. Queue length is overestimated under penetration rates of 5% and 15% because these estimation results show more queues formed on freeway. Queue length is underestimated under penetration rates of 25%, 30% and 35% because these results show that the queue only forms for one or two times. Overall, under the a penetration rate of 20% and 40%, the queue length estimation is more accurate than other penetration rates.

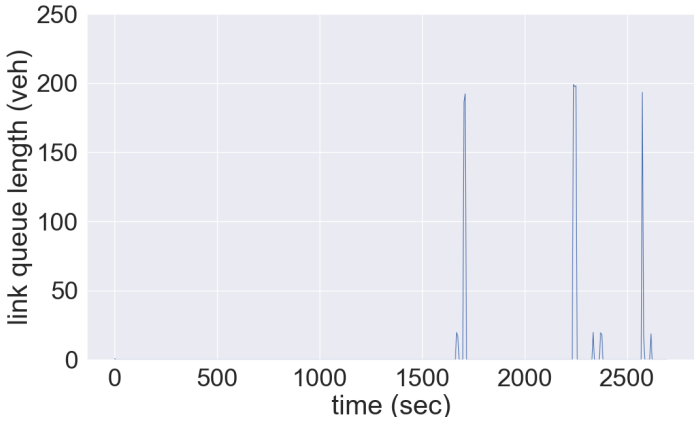
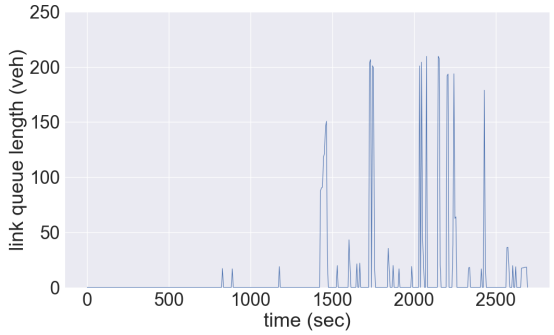
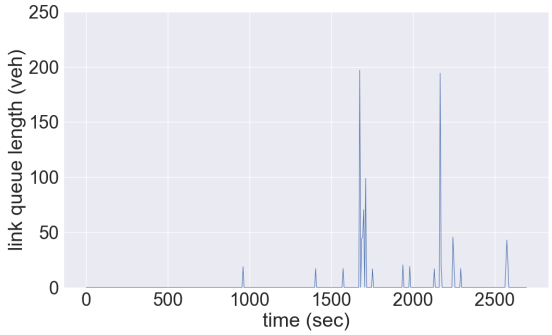


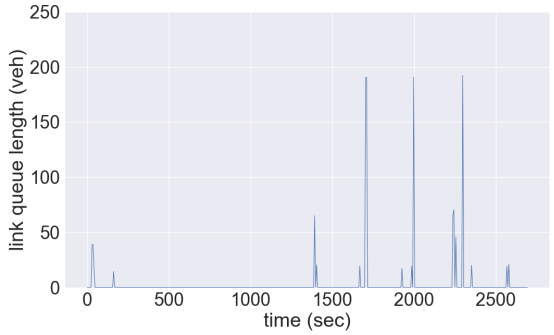
Figure. 78: Actual link queue length in congested condition (100% vehicles)



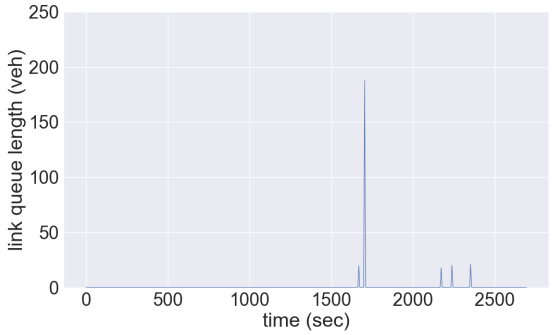
(a) 5% penetration



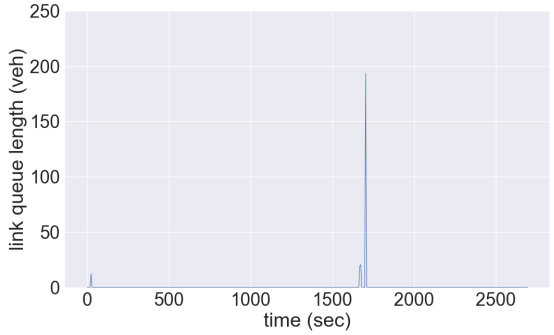
(b) 10% penetration



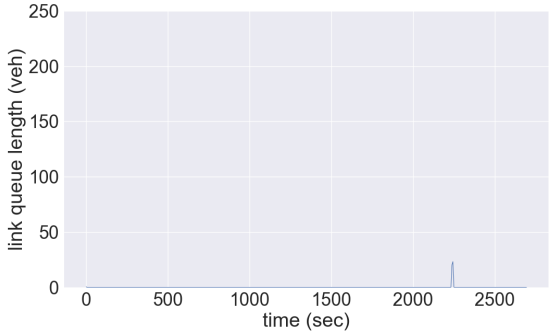
(c) 15% penetration



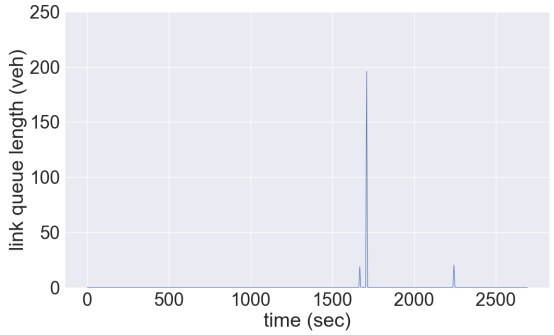
(d) 20% penetration



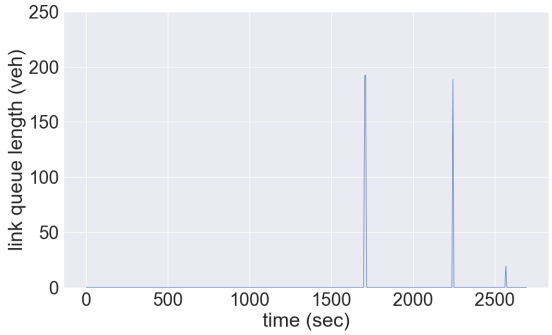
(e) 25% penetration



(f) 30% penetration



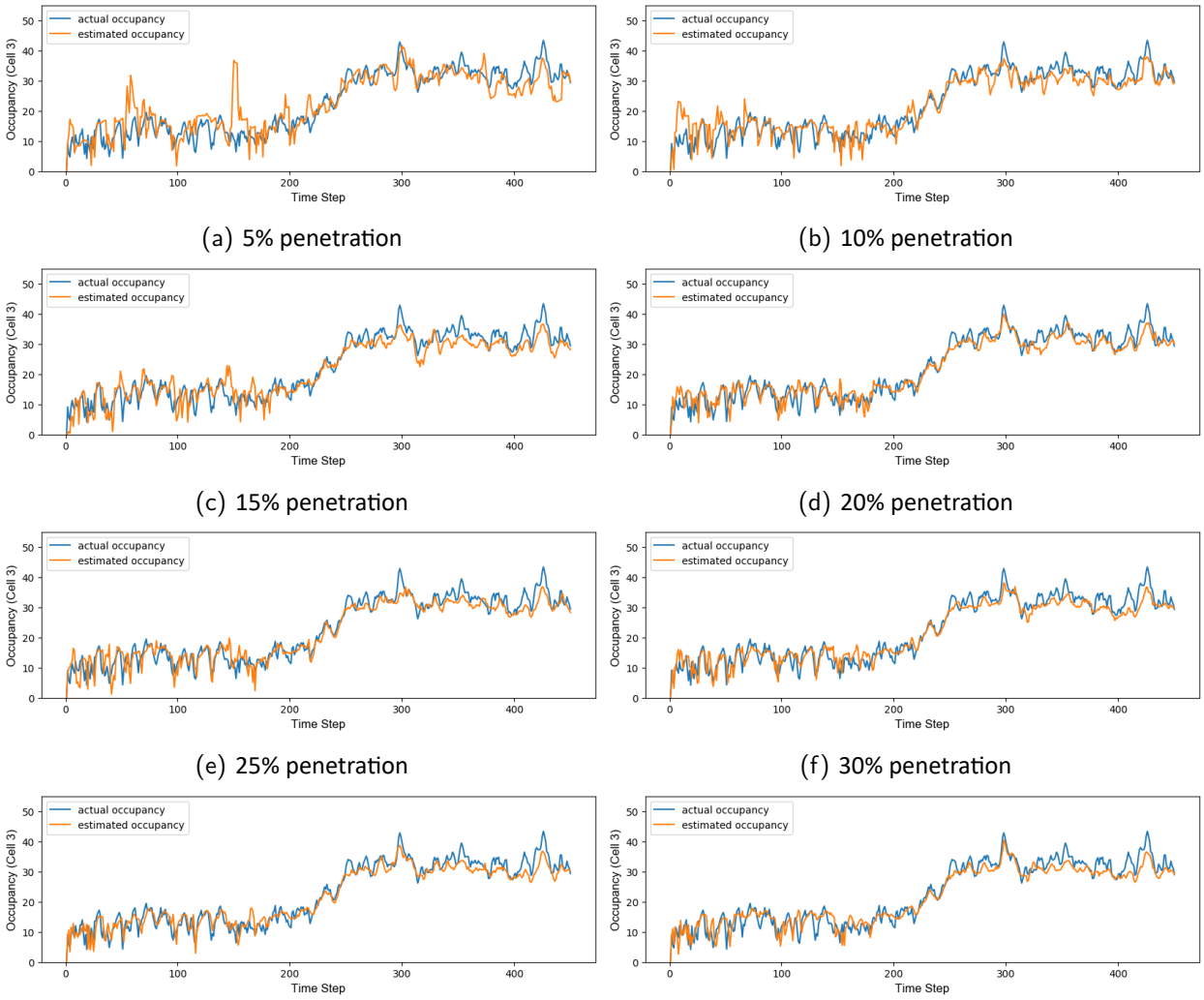
(g) 35% penetration



(h) 40% penetration

Figure. 79: Queue length estimation with BSMs from AIMSUN (congested condition)

9.3.7 The effect of increasing penetration rates



(a) 5% penetration

(b) 10% penetration

(c) 15% penetration

(d) 20% penetration

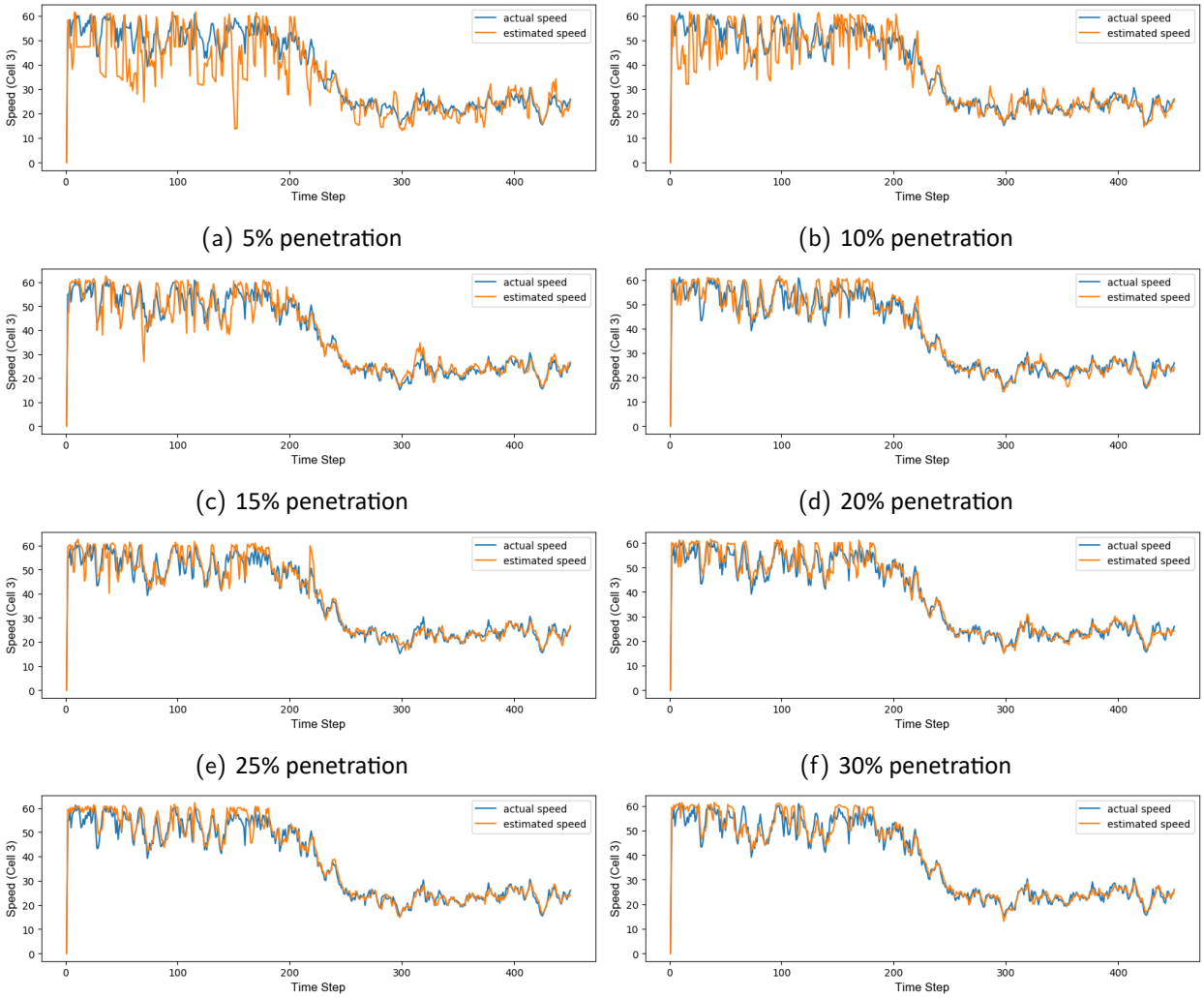
(e) 25% penetration

(f) 30% penetration

(g) 35% penetration

(h) 40% penetration

Figure. 80: Effect of penetration rates to density estimation for cell 3 (congested condition)



(a) 5% penetration

(b) 10% penetration

(c) 15% penetration

(d) 20% penetration

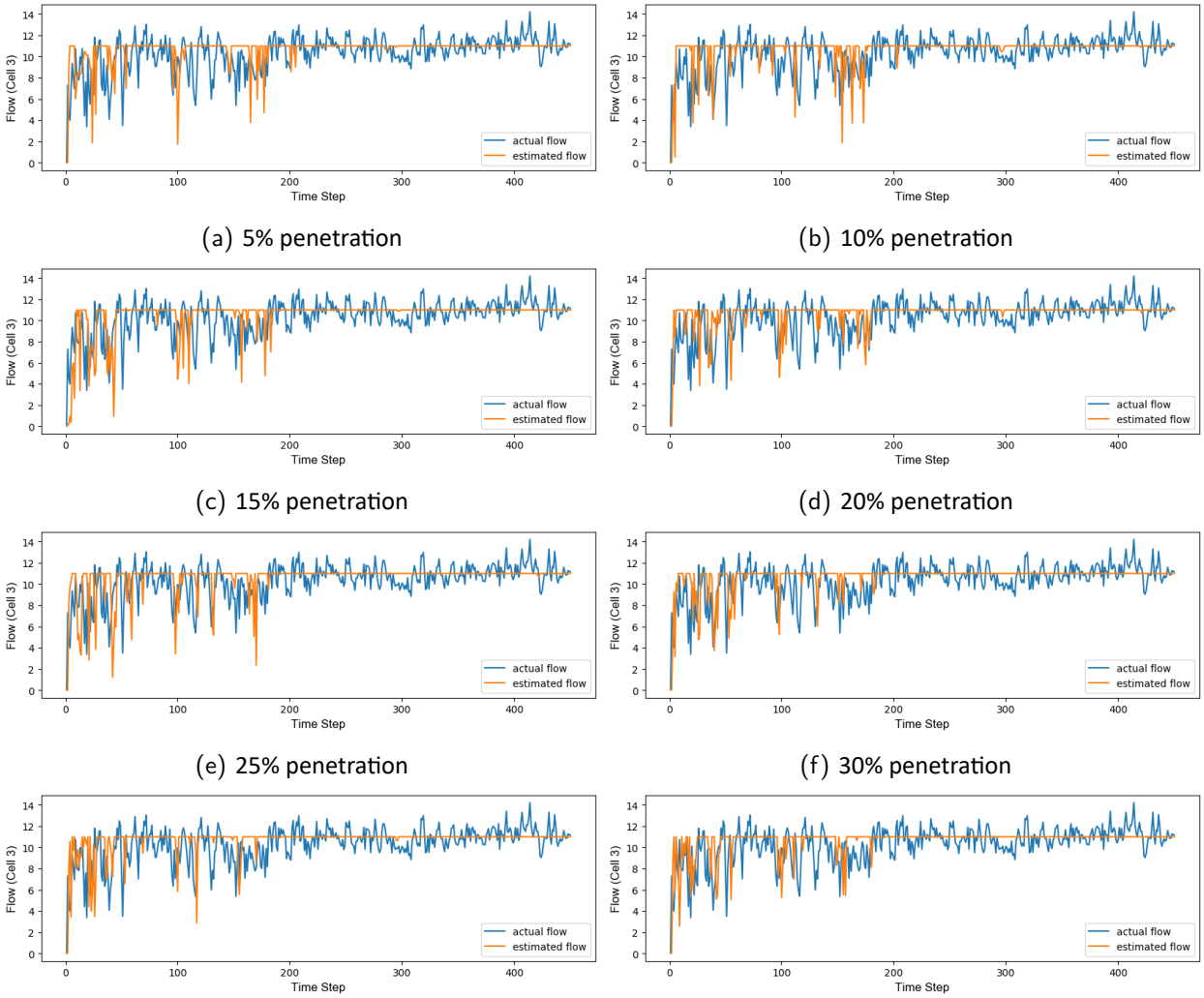
(e) 25% penetration

(f) 30% penetration

(g) 35% penetration

(h) 40% penetration

Figure. 81: Effect of penetration rates to speed estimation for cell 3 (congested condition)



(a) 5% penetration

(b) 10% penetration

(c) 15% penetration

(d) 20% penetration

(e) 25% penetration

(f) 30% penetration

(g) 35% penetration

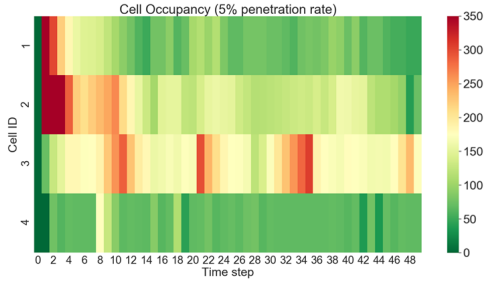
(h) 40% penetration

Figure. 82: Effect of penetration rates to flow estimation for cell 3 (congested condition)

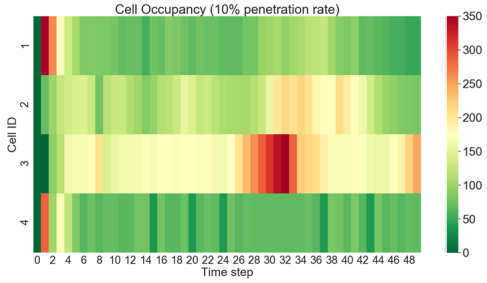
In Section 9.3.7, the estimated density, speed, and flow for one cell (cell 3) is compared to their actual values, which is shown is Figures 80, 81, and 82. When the penetration rate gets larger, the fluctuation of the curve reduces. This phenomenon is obvious for density and speed estimation but not for flow estimation.

9.4 BSMs from radar data

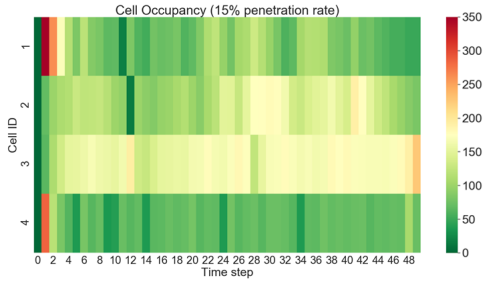
9.4.1 Density estimation



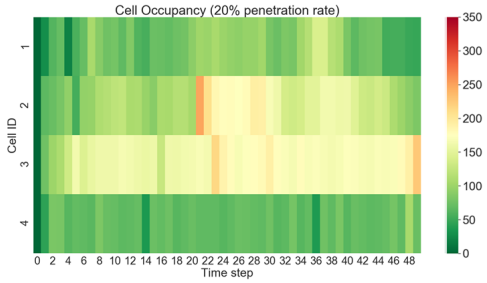
(a) 5% penetration



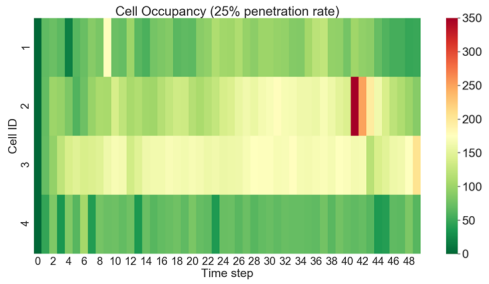
(b) 10% penetration



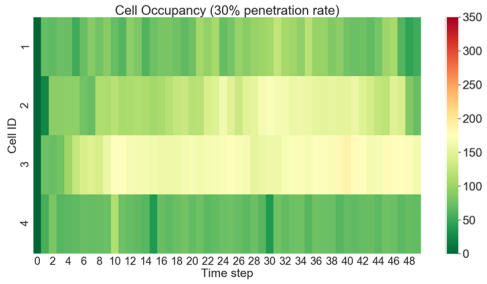
(c) 15% penetration



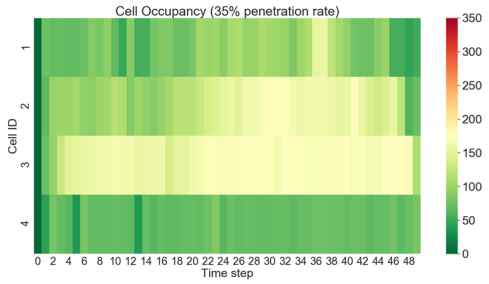
(d) 20% penetration



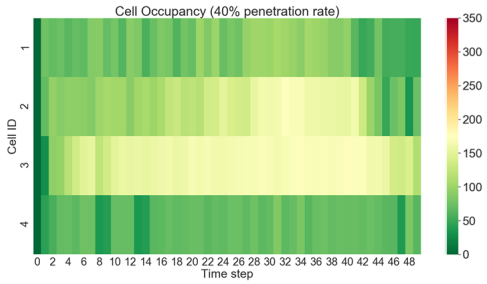
(e) 25% penetration



(f) 30% penetration



(g) 35% penetration



(h) 40% penetration

Figure. 83: Density estimation with BSMs from radar data

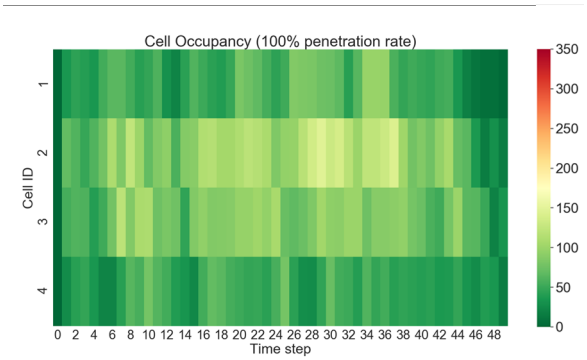


Figure. 84: Actual density with BSMs from radar data (100% vehicles)

9.4.2 Speed estimation

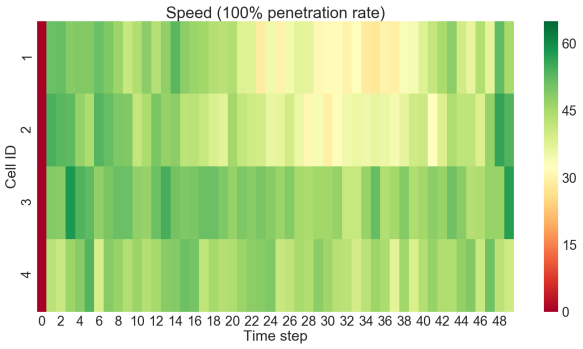
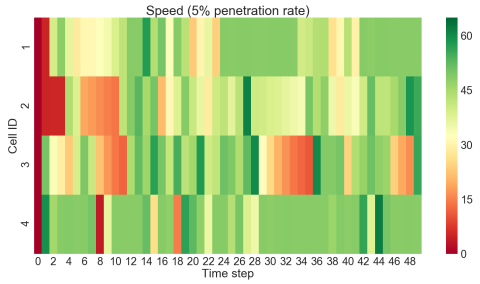
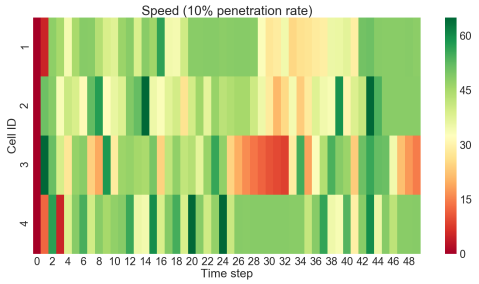


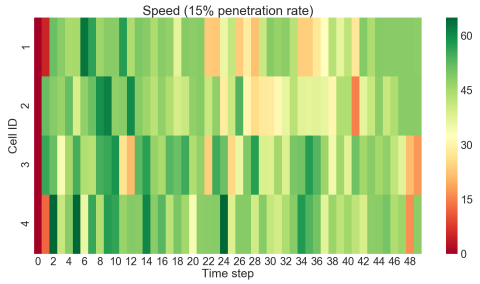
Figure. 85: Actual speed with BSMs from radar data (100% vehicles)



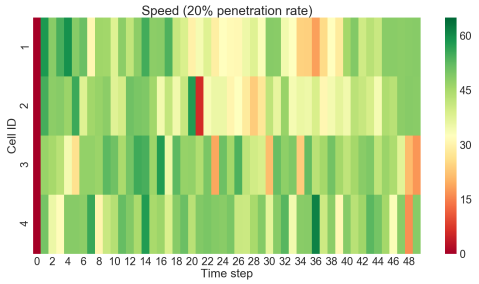
(a) 5% penetration



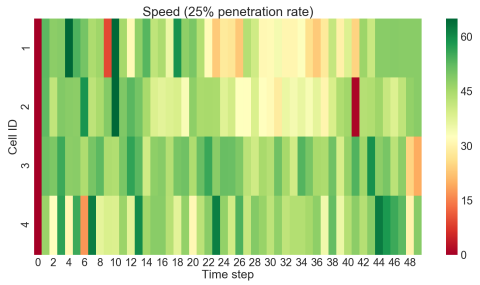
(b) 10% penetration



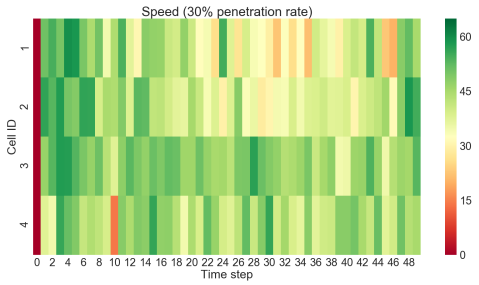
(c) 15% penetration



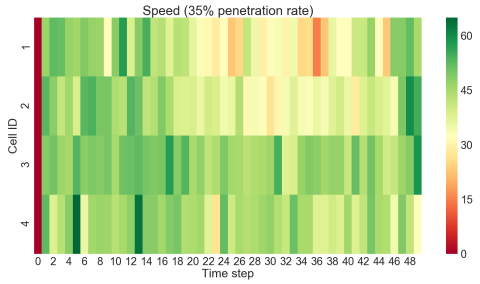
(d) 20% penetration



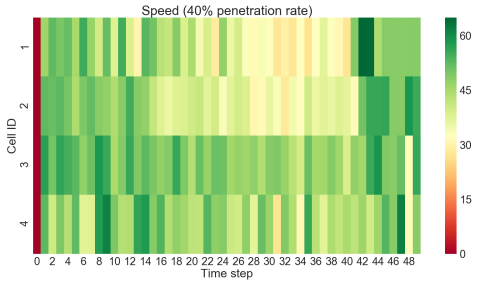
(e) 25% penetration



(f) 30% penetration



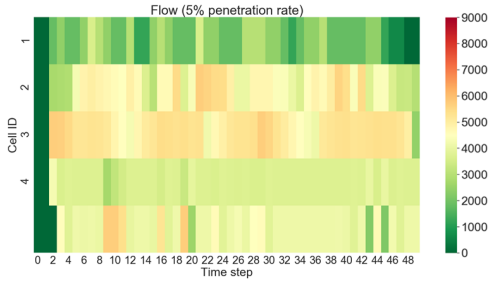
(g) 35% penetration



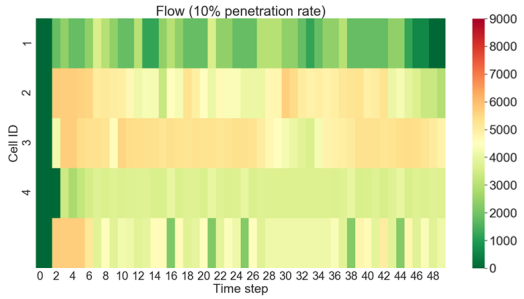
(h) 40% penetration

Figure. 86: Speed estimation with BSMs from radar data

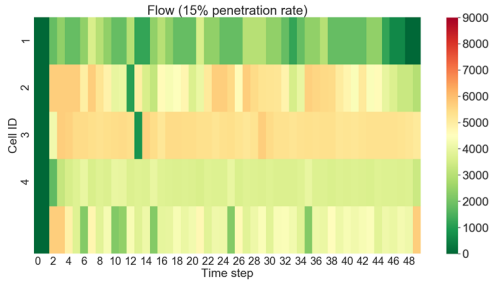
9.4.3 Flow estimation



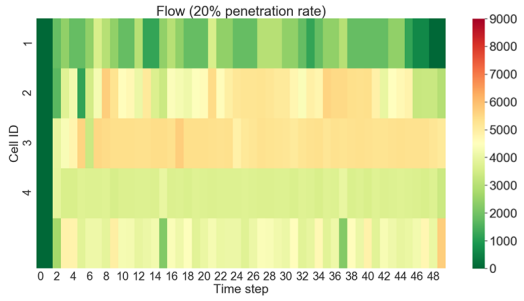
(a) 5% penetration



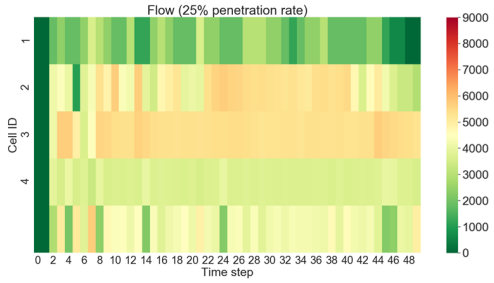
(b) 10% penetration



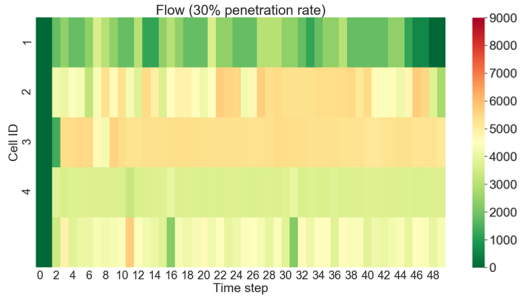
(c) 15% penetration



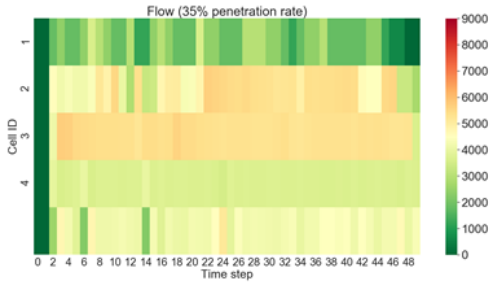
(d) 20% penetration



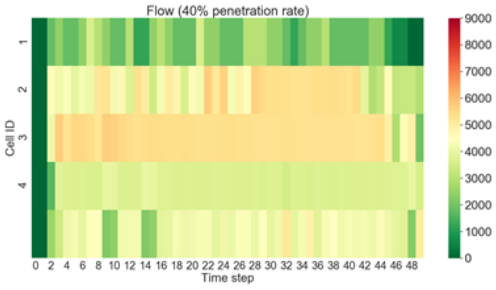
(e) 25% penetration



(f) 30% penetration



(g) 35% penetration



(h) 40% penetration

Figure. 87: Flow estimation with BSMs from radar data

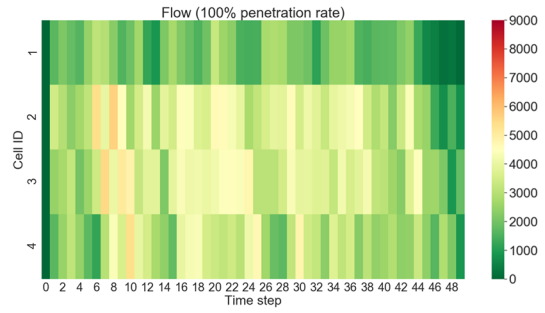
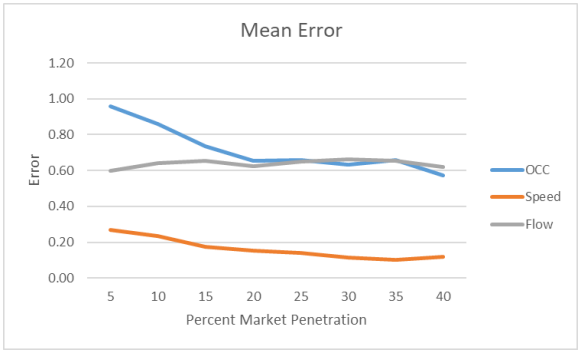
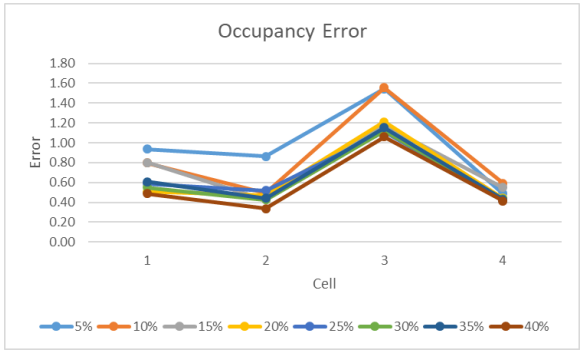


Figure. 88: Actual flow with BSMs from radar data (100% vehicles)

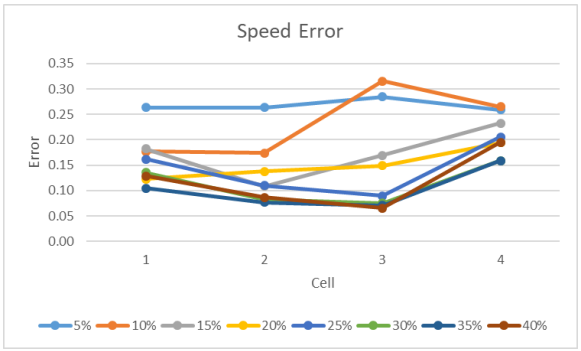
9.4.4 Estimation error



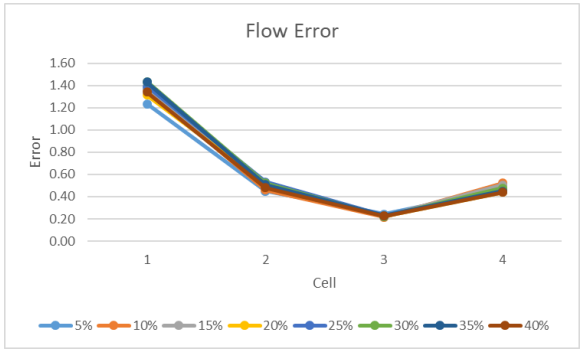
(a) Overall error



(b) Density error



(c) Speed error



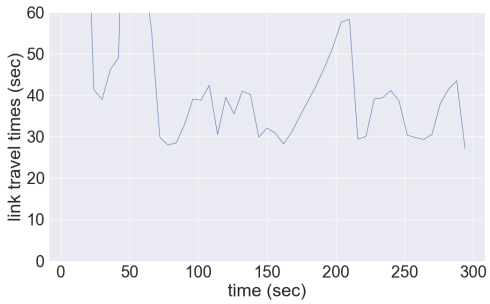
(d) Flow error

Figure. 89: Estimation errors with BSMs from radar data

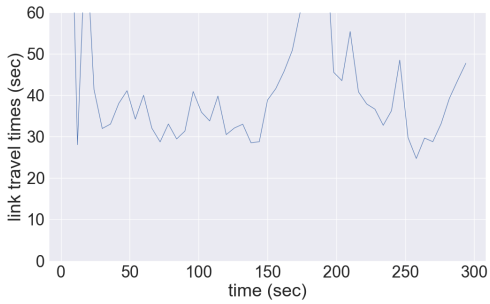
The overall shapes and values of the charts seen in Figure 89 from the Task 4 radar data error are very similar to those seen in the Task 3, but in some cases the error is larger in Task 4 than it is in Task 3. The density error still has a spike at cell 3, with a range of values between 100% and 160%

error. This is less accurate than in Task 3, which had a range of 90% to 130% error. This is reflected in the mean error in which the mean density error in Task 4 is between 4 and 10 percentage points greater than it was in Task 3, depending on the market penetration. The speed error seems to be relatively unchanged from Task 3 to Task 4, although the mean speed error may have gotten a percentage point or two worse in Task 4. The flow error is less accurate in Task 4, as seen in cell 1 where the error ranges from 121% to 143% compared to cell 1 in Task 3 which ranges from 80% to 100% error. The mean flow error in Task 4 is between 5 and 10 percentage points greater than in Task 3. Normally, the estimation error for the first cell will drop if correct values of input flow is provided. As the estimation error for the first cell increases compared to the estimation error in Task 3, it can be concluded that the calculated values from vehicle trajectory information may have some errors.

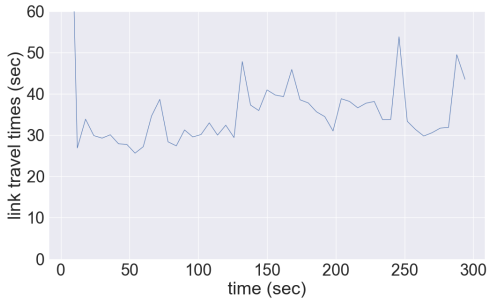
9.4.5 Travel time estimation



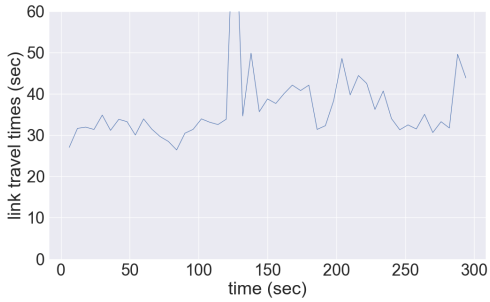
(a) 5% penetration



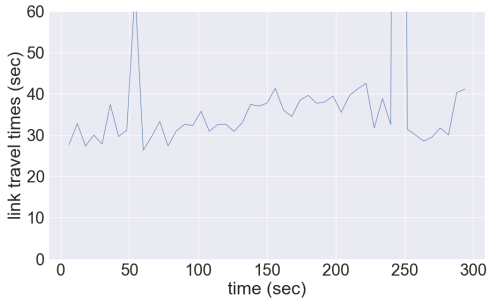
(b) 10% penetration



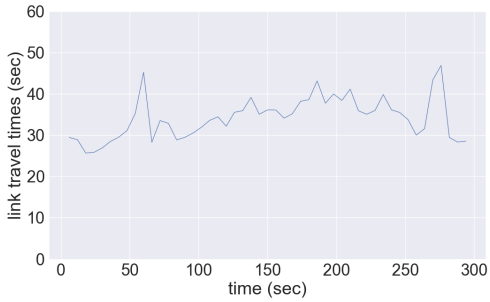
(c) 15% penetration



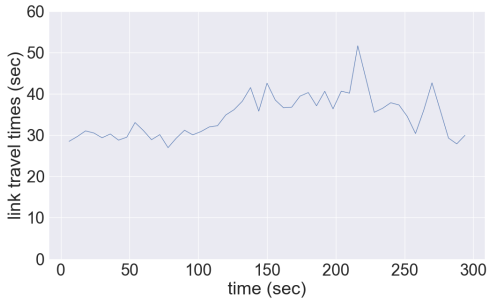
(d) 20% penetration



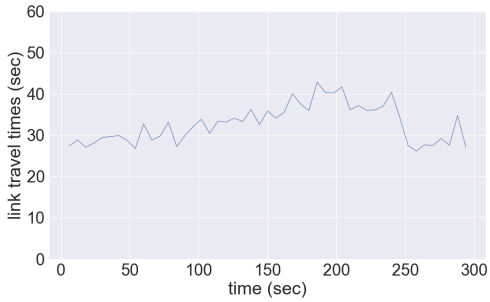
(e) 25% penetration



(f) 30% penetration



(g) 35% penetration



(h) 40% penetration

Figure. 90: Travel time estimation with BSMs from radar data

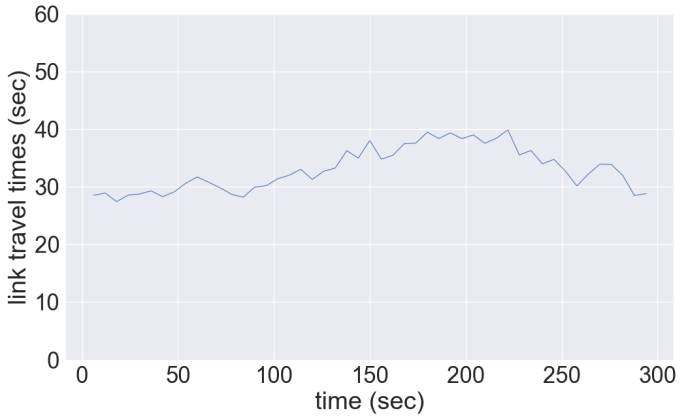


Figure. 91: Actual link travel time (100% vehicles)

Figure 90 shows estimated travel time. When the penetration rate is larger than 25%, the travel time can be accurately estimated. When the penetration rate is small, the model tends to over-estimate the travel time which indicates that the speed is underestimated when the penetration rate is small.

9.4.6 Queue length estimation

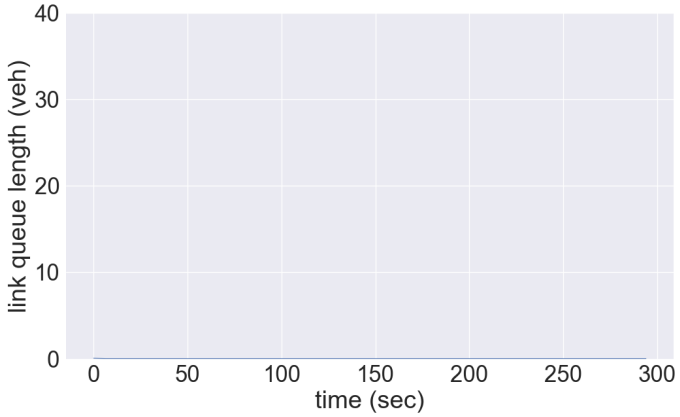
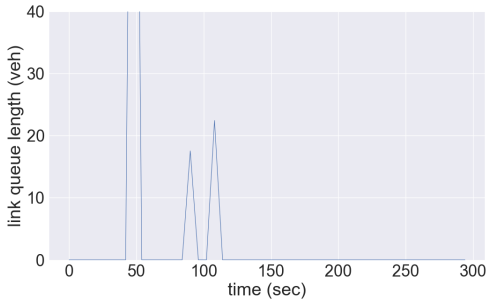
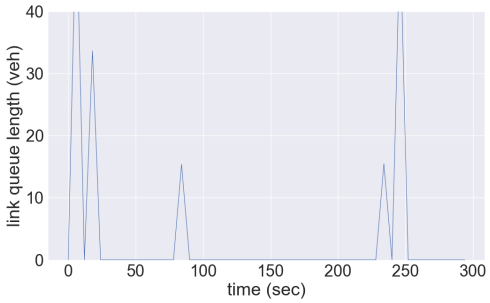


Figure. 92: Actual link queue length (100% vehicles)

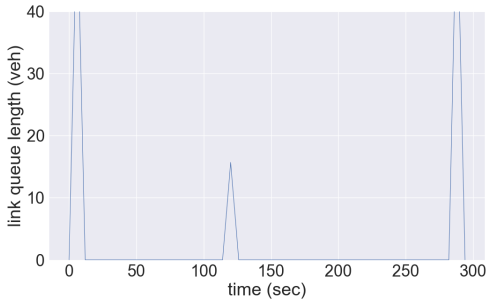
Figure 92 shows the actual value for queue length. As the road is not congested, there is no queue formed. However, all tests overestimated the queue length as short queues are formed on road two or three times based on the estimation results.



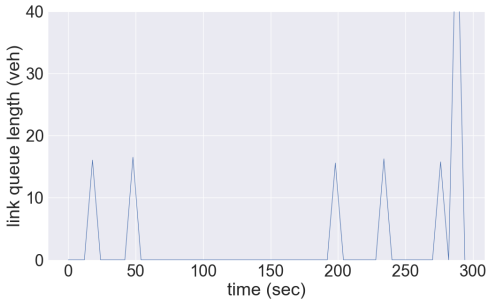
(a) 5% penetration



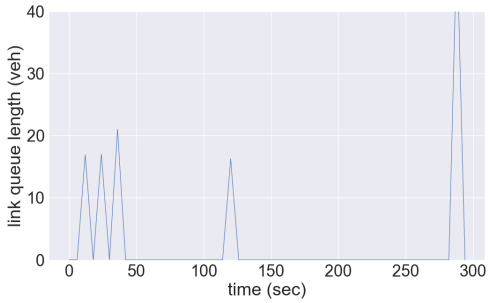
(b) 10% penetration



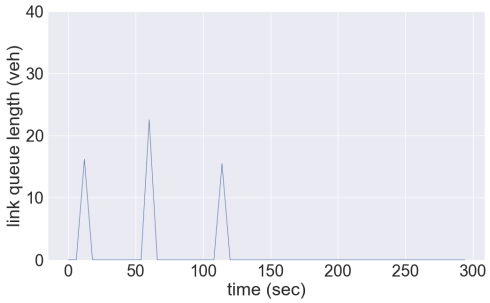
(c) 15% penetration



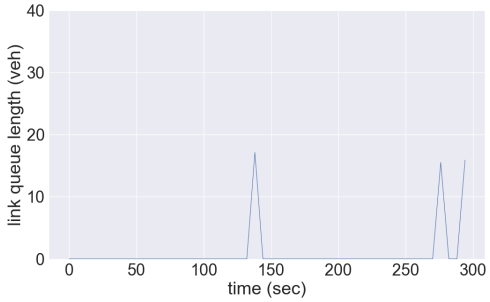
(d) 20% penetration



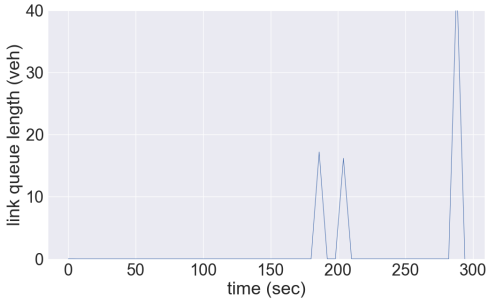
(e) 25% penetration



(f) 30% penetration



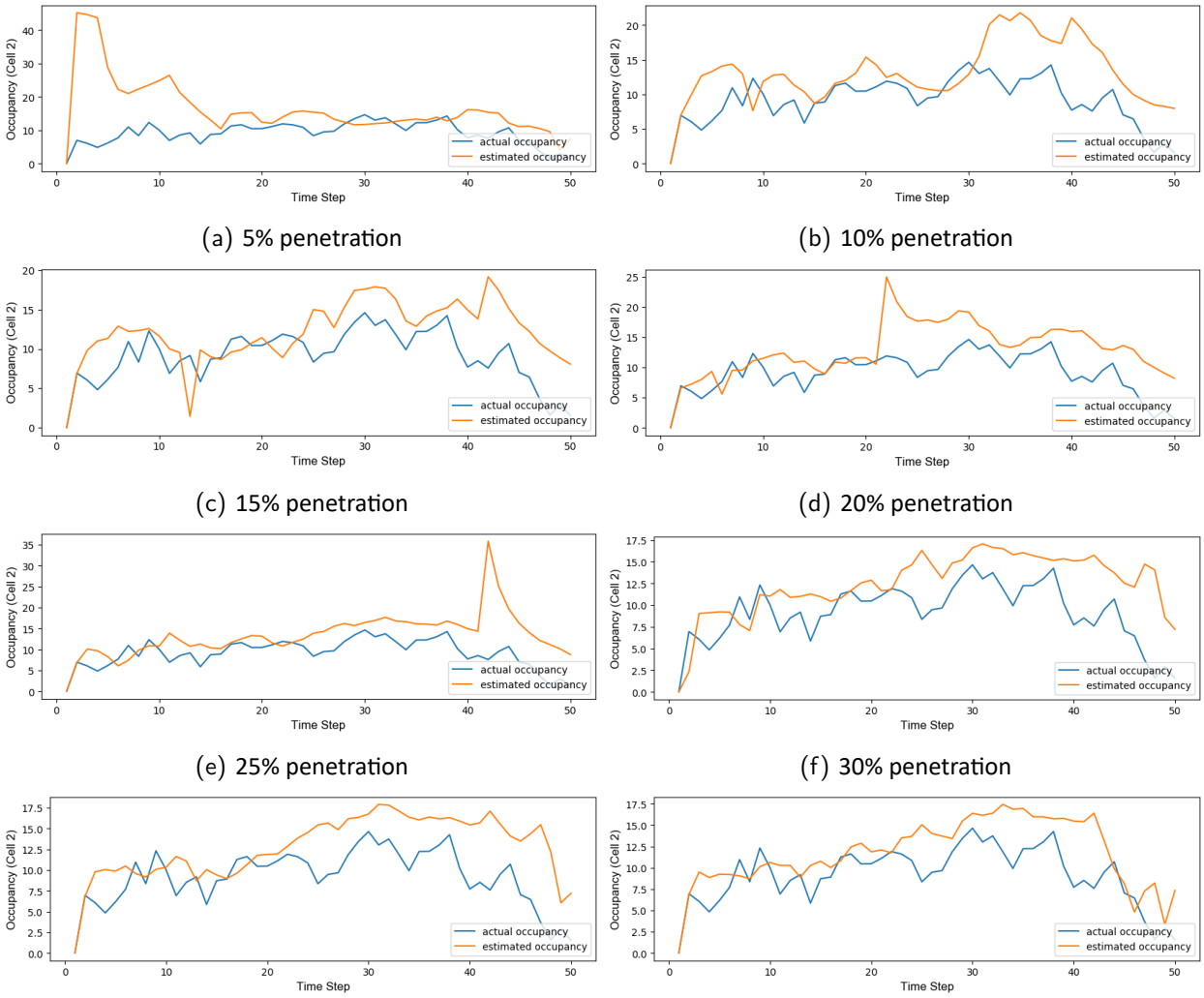
(g) 35% penetration



(h) 40% penetration

Figure. 93: Queue length estimation with BSMs from radar data

9.4.7 The effect of increasing penetration rates



(a) 5% penetration

(b) 10% penetration

(c) 15% penetration

(d) 20% penetration

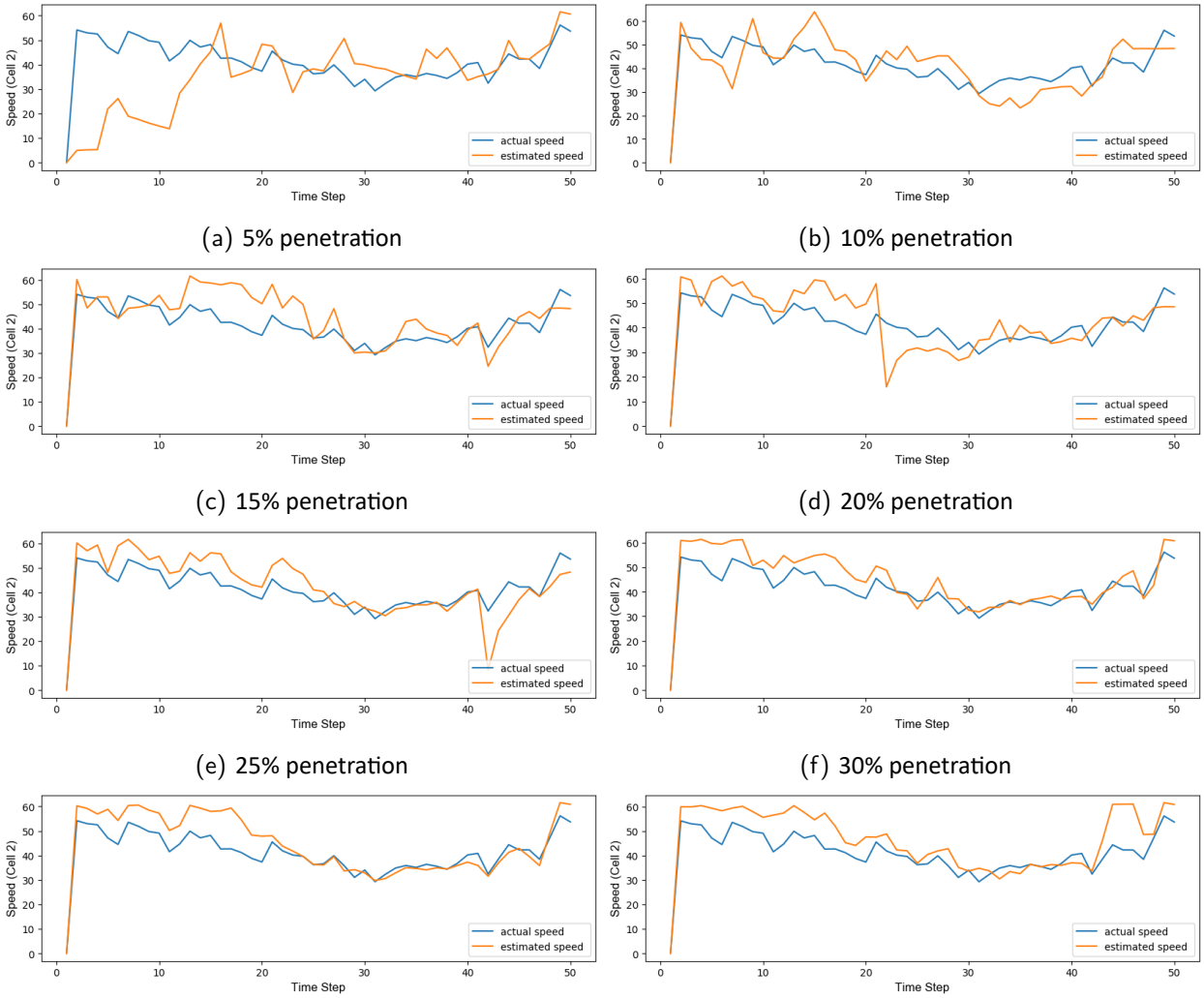
(e) 25% penetration

(f) 30% penetration

(g) 35% penetration

(h) 40% penetration

Figure. 94: Effect of penetration rates to density estimation for cell 2



(a) 5% penetration

(b) 10% penetration

(c) 15% penetration

(d) 20% penetration

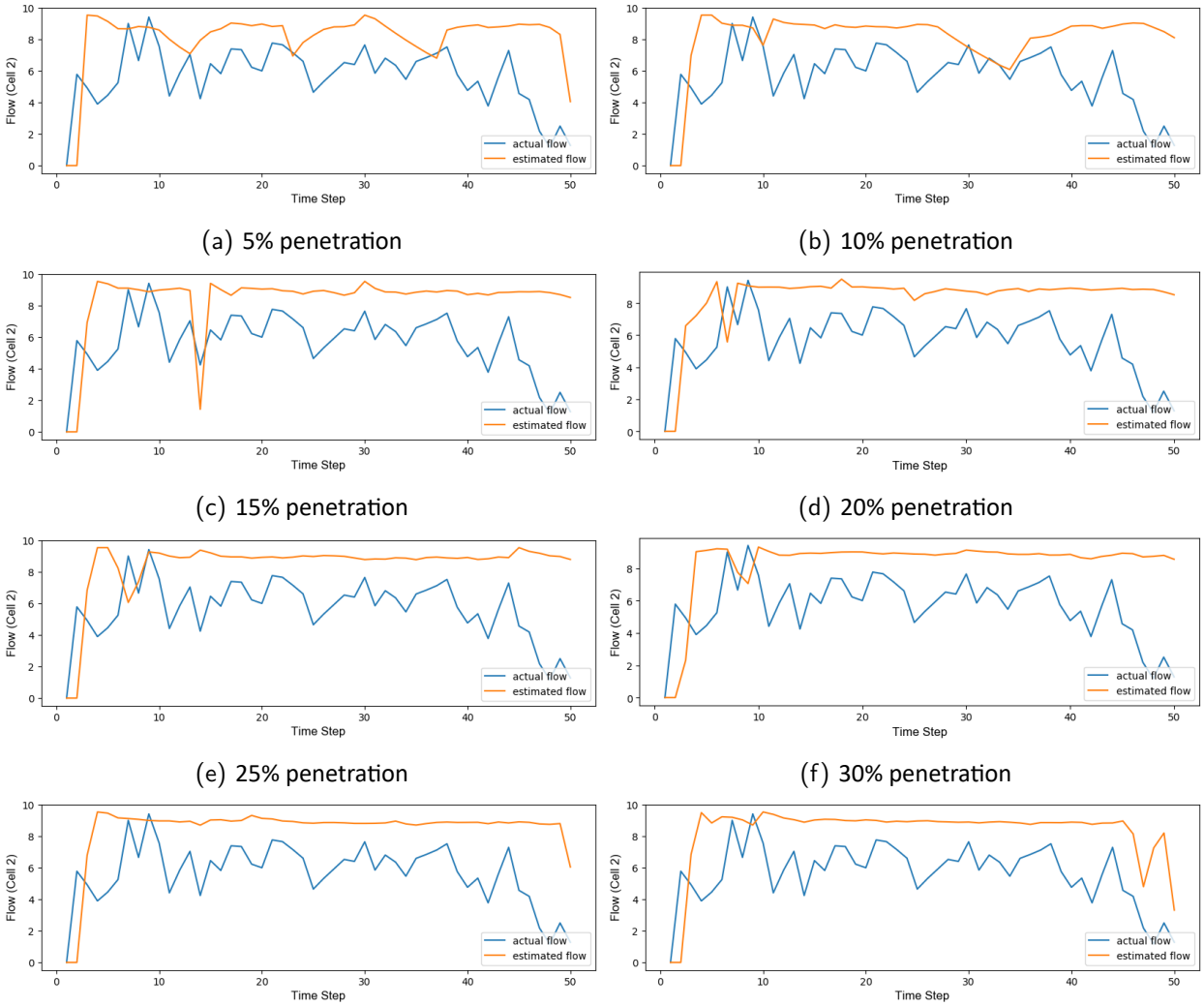
(e) 25% penetration

(f) 30% penetration

(g) 35% penetration

(h) 40% penetration

Figure. 95: Effect of penetration rates to speed estimation for cell 2



(a) 5% penetration

(b) 10% penetration

(c) 15% penetration

(d) 20% penetration

(e) 25% penetration

(f) 30% penetration

(g) 35% penetration

(h) 40% penetration

Figure. 96: Effect of penetration rates to flow estimation for cell 2

9.5 Discussion

This section updates the model used for the last section by providing measurements for input flow to the target road section. The accuracy of the updated model are evaluated using BSMs from AIMSUN under both congested and uncongested conditions and BSMs prepared using radar data. Input flows for target road sections in AIMSUN are extracted by adding loop detectors at one end of the road before running the simulation. Input flows for the experiment using BSMs from radar data are extracted by processing the vehicle trajectory information.

For the experiment using BSMs generated by AIMSUN, errors for density, speed, and flow are reduced by adding measurements for input flows compared to the estimation result in Task 3. The accuracy for travel time and queue length estimations are not changed compared to the estimation result in Task 3.

In Task 4, the accuracy of the Kalman filter is much lower when applied to BSMs from radar data. The accuracy also drops when adding measurements of input flows to the model when comparing the estimation results in Task 3 and Task 4. One possible reason is that BSM data generated from radar data may have some errors which make the extracted values for input flow incorrect.

10 Experiments on a Large-scale Network with a Traffic Monitoring System

In this task, the project team create a traffic monitoring system that is capable of estimating traffic states using basic safety messages (BSMs) from a large traffic network. The efficiency of the system is tested using BSMs generated by a microscopic simulation model.

10.1 BSM traffic monitoring system

10.1.1 System structure

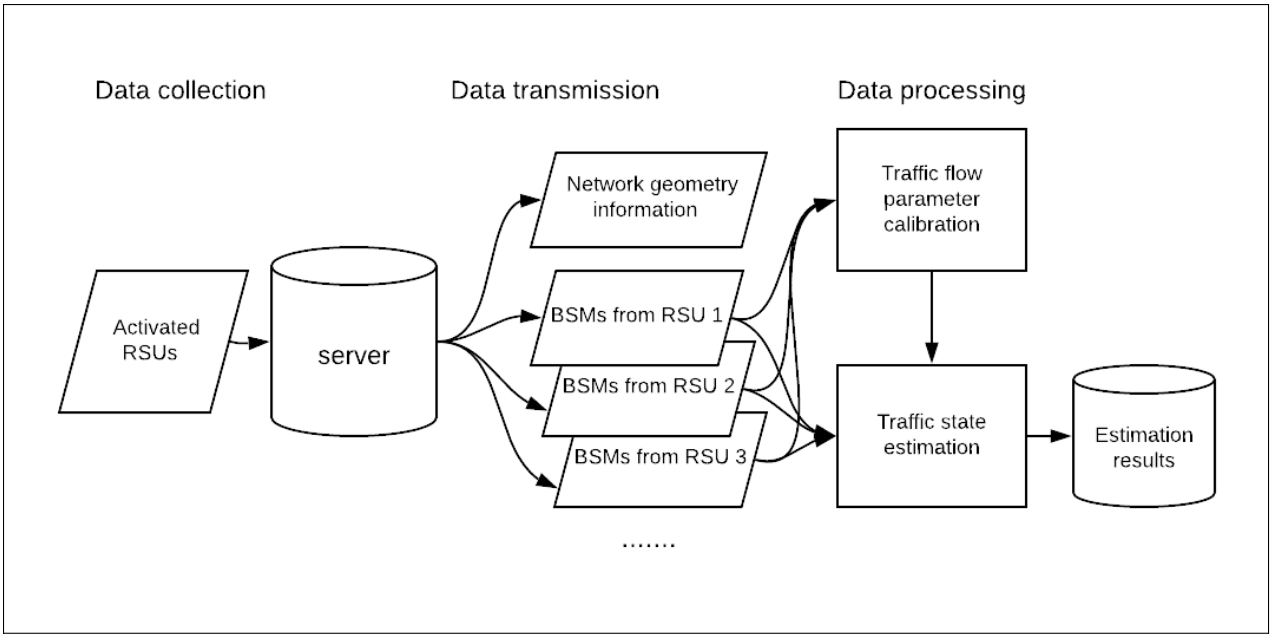


Figure. 97: System structure

As shown in Figure 97, the traffic monitoring system has three components: the data collection module, and the data transmission module, and the data processing module. The data collection module has multiple road-side units (RSUs) covering major roads of the traffic network. RSUs collect BSMs from equipped vehicles every 0.1 sec. BSMs collected by RSUs are then uploaded to an online server, which uses "Postgres" as its main database management system. On the server, each database includes all BSMs collected from a network with a fixed penetration rate. Each table in a database stores BSMs collected by an RSU. The database also includes the geometry information of road sections covered by each RSU, such as lane number, road length, on-ramp number, and off-ramp number. The server enables all computers to access the database remotely with granted account, password, and IP address. In the data processing module, traffic states of

target road sections are estimated by one or multiple computers. After the traffic state estimation, the results are stored in several local databases.

10.1.2 BSMs generation environment

BSMs for this task is generated by a calibrated microscopic simulation model in AIMSUN. Figure 98 shows the road network included in the simulation model. This network covers the city of Richfield, and part of the cities of Bloomington, Edina and Minneapolis. Freeway I-35w and I-494, and highway 62 and 77 are also included in the model. There are 69 road-side units (RSUs) set

Figure. 98: Target test area

in the network, and each RSU is capable to collect BSMs for equipped vehicles from several road sections. Figure 99 shows an example of the detection range of two RSUs represented by the

orange box and the yellow box respectively. BSMs from vehicles on sections 1 and 2 are collected by RSU 1 and BSMs from vehicles on sections 3, 4, 5, and 6 are collected by RSU 2. These sections have different lane numbers and connect to different number of on-ramps or an off-ramps. For example, section 1 in Figure 99 has four lanes but section 2 has five lanes. Section 2 has an off ramp but section 1 is merged with an on ramp.

10.1.3 Kalman filter generation

In the detection range of an RSU, two road segments with different lane numbers are considered as two RSU sections. To initialize a Kalman filter, the geometry information of an RSU section, such as lane number and section length, is needed. The time step size Δt is set to be 6 sec in this project, which indicates that the Kalman filter estimates traffic states every 6 seconds. A smaller time step size can result in a larger computation time. In a Kalman filter, the road section is divided into several cells, which is the basic unit for traffic state estimation. The cell length l is calculated by the following equation:

$$l \geq u_{\mathrm{f}}^{0} \Delta t \tag{33}$$

where u_{f}^{0} is the free flow speed. The cell length is usually equal to the time step size Δt times the initial value of the free-flow speed u_{f}^{0}, then the number of cells for an RSU section can be calculated using the section length L divided by the cell length l. If the remainder of L/l is too small, which violates inequality (33), the last cell in the RSU section will use a larger cell length. For example, if the cell length is 600 ft, and the section length is 1300, then the first cell uses a cell length of 600 ft and the second cell use a cell length of 700 ft.

For two connected RSU sections that belong to the same RSU detector range, the estimated flow going out of the upstream section is used to calculate the input flow of the downstream section, as shown in both Figures 100 and 101. Moreover, if a road or several connected roads are covered by multiple RSUs, then these RSUs can form an RSU group, as shown in Figure 100. The estimated flow going out of last RSU section covered by the upstream RSU can be used to calculate the input flow to the first RSU section covered by the downstream RSU. If the detection ranges of two RSUs are not connected, then these two RSUs still belong to different RSU groups, as shown in Figure 101. Before the start of traffic state estimation, the traffic monitoring system can generate a list of RSU groups given the list of activated RSUs and network geometry information.

10.2 Experiments with the traffic monitoring system

The efficiency of the system was tested in two scenarios under different penetration rates. The two scenarios evaluate the computational requirements for combining different numbers of RSUs

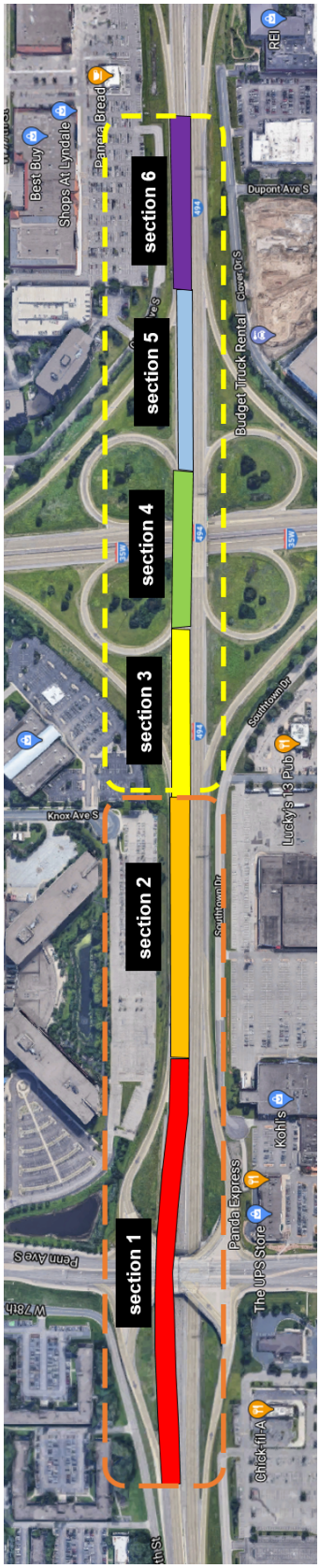


Figure. 99: An example of RSUs

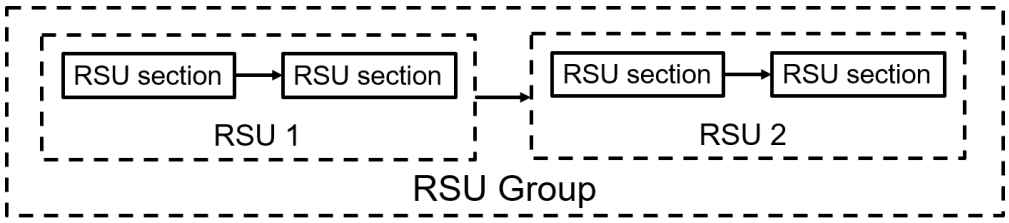


Figure. 100: Examples of connected RSU

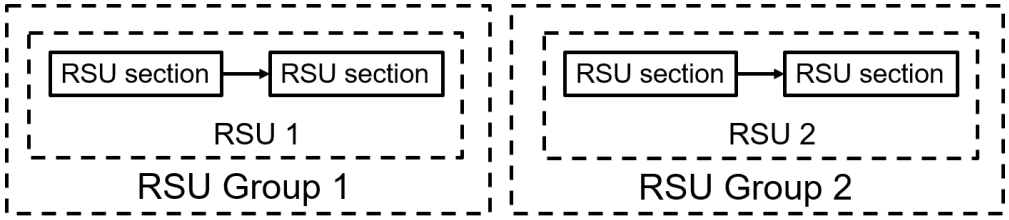


Figure. 101: Examples of separated RSU groups

as input data for one traffic state estimator, at different market penetrations. Using multiple RSUs as input to a single traffic state estimator results in modeling a larger section of road and also increases the number of data points input per time step. However, as seen in Tasks 3 and 4, much of the error in the traffic state estimation occurs at the start of the road segment because limited data is available on input flows. Combining RSUs to model a larger segment of road could therefore reduce the error at the cost of requiring additional input data (and correspondingly higher computation times). Higher market penetrations also increase computation times due to providing more input data.

10.2.1 Simulated BSM data

The Minnesota Traffic Observatory (MTO) constructed an Aimsun model of the freeway network in the Minneapolis/St. Paul metropolitan region. The original plan was to add RSUs that covered the entire freeway network and conduct traffic state estimation. Unfortunately, due to the large computation time, as well as limited time by engineering personnel, only 10 RSUs were implemented for data collection. Data was collected for 12,600 seconds (3.5 hours) of simulation. We have results from scenarios including 10 RSUs combined into one traffic state estimation (the worst case for computation time) and each RSU individually.

10.2.2 Experiment scenarios

In the first scenario, there are 10 RSUs activated in the experiment (shown in Table 11). Most RSUs are located on freeway ramps that connect two main directions of freeways. All these RSUs are

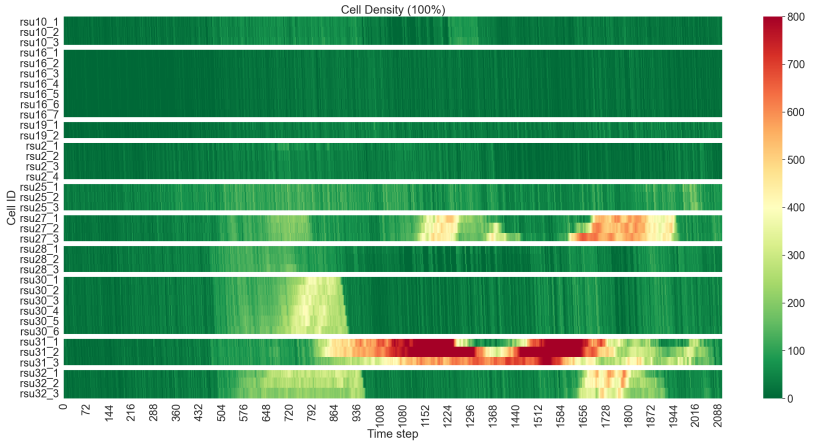
isolated from each other, so there are 10 RSU groups created in this scenario and each RSU group only includes one RSU. The second scenario activates 10 RSUs on freeway I–35 (shown in Table 12). These RSUs are connected and they form one RSU group covering the entire freeway section. For each scenario, there are four penetration rates test: 20%, 40%, 60%, 80&. A penetration rate only means a proportion of vehicles in the entire network are equipped vehicles, instead of making the proportion of equipped vehicles in target road sections equal to the penetration rate. For example, a penetration rate of 20% does not guarantee that 20% of vehicles on the detection section of RSU 10 are equipped.

Table 11: Activated RSUs in scenario 1

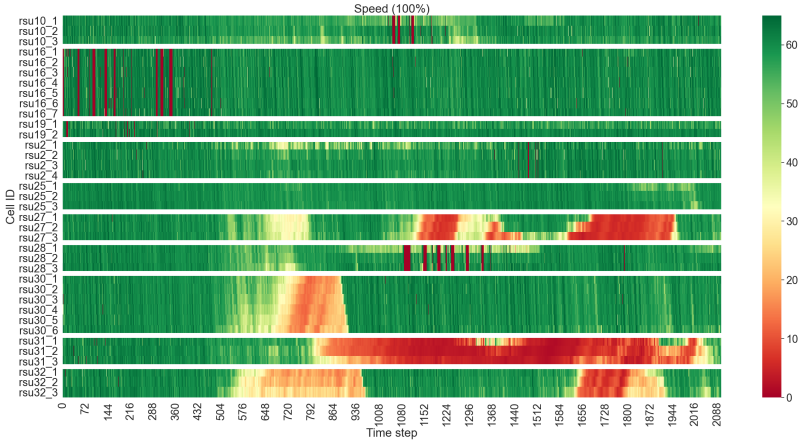
RSU name	cell number	total length (ft)
rsu10_62_WB	3	1992.64
rsu16_77_SB	7	4123.45
rsu19_77_SB	2	1712.61
rsu2_35_NB	4	2447.68
rsu25_494_WB	3	2181.38
rsu27_494_WB	3	2046.76
rsu28_494_EB	3	2096.03
rsu30_494_EB	6	3956.48
rsu31_494_WB	3	604.46
rsu32_494_EB	3	2197.61
Total	37	23359.1

Table 12: Activated RSUs in scenario 2

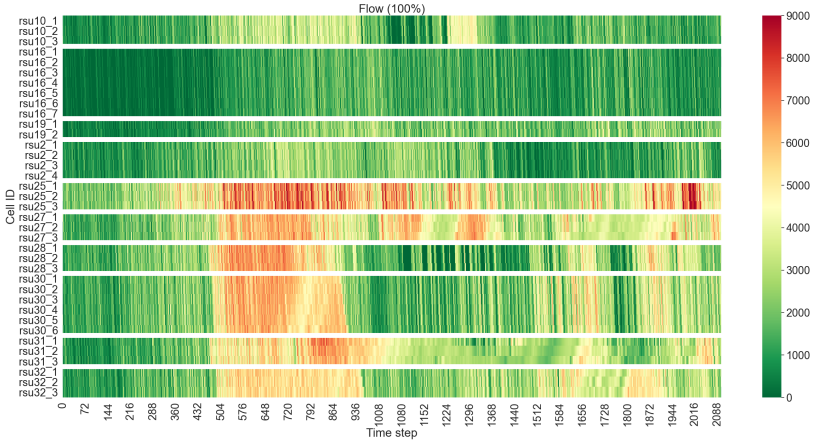
RSU name	cell number	total length (ft)
rsu40_35_NB	3	2022.7
rsu68_35_NB	4	1919.45
rsu37_35_NB	8	4651.14
rsu67_35_NB	3	1329.47
rsu2_35_NB	4	2447.68
rsu66_35_NB	2	1332.39
rsu65_35_NB	6	4276.83
rsu43_35_NB	3	1941.04
rsu5_35_NB	4	2850.98
rsu62_35_NB	5	2915.1
Total	42	25686.78



(a) 100% densities in scenario 1

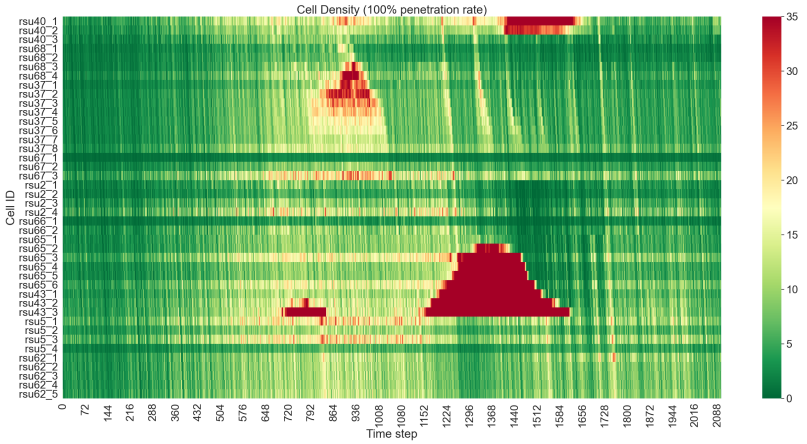


(b) 100% speeds in scenario 1

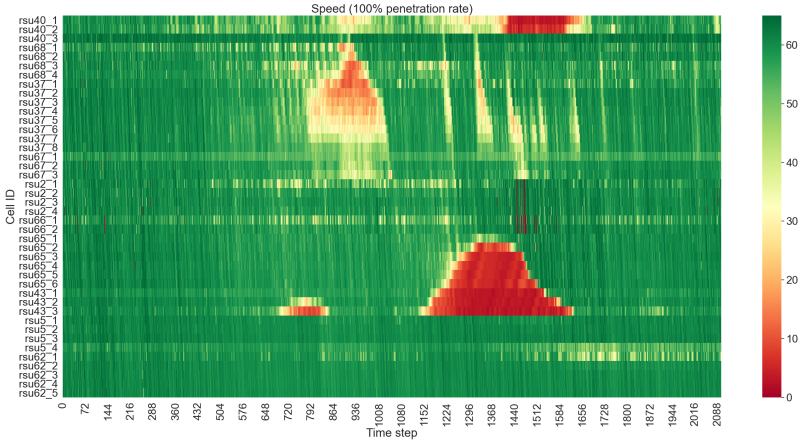


(c) 100% flows in scenario 1

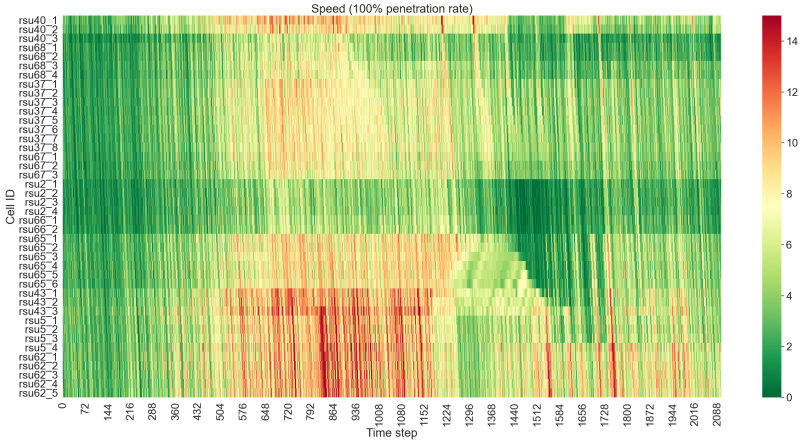
Figure. 102: Actual traffic states for scenario 1



(a) 100% densities in scenario 2



(b) 100% speeds in scenario 2



(c) 100% flows in scenario 2

Figure. 103: Actual traffic states for scenario 2

10.2.3 Ground truth database preparation

The ground truth database needs to be prepared in advance to compare with the estimated results based on equations mentioned in section 6.1. Computation times used for extracting traffic states from 100% BSMs using Edie's method for first scenario and the second scenario are 4 hours and 8 hours respectively.

Figures 102a, 102b, and 102c show the actual traffic states (density, speed, and flow) detected by all ten RSUs in scenario 1. Red color represents traffic congestion. The road segments covered by the first few RSUs are not congested condition while the road segments covered by the last two RSUs are congested. Figures 103a, 103b, and 103c show the actual traffic states in scenario 2.

Figures 103a, 103b, and 103c show the actual traffic states detected by ten RSUs in scenario 2. At the bottom of these figures, we can observe the propagation of the traffic congestion from the detection area of RSU 43 to RSU 65. This traffic congestion generates at time step 1235 and disappears at time step 1635, which lasts 40 minutes.

10.2.4 Computation time

One important objective of this task is to test if the traffic monitoring system is able to operate in real time, when it is implemented in a large scale network. If one computer cannot run the system in real-time, what is a reasonable number of computers that can make it run in real-time. To determine if this system can run in real-time, we can compare the computation time with the simulation time, or compare the computation time in each time step with time step size (6 sec in this study). The computation time for this system is mainly composed of the time used for running SQL query and for running the Kalman filter to get estimated traffic states. For this task, we used computers with Intel Xeon processor (3.60 GHz) with 32 GB RAM.

Table 13 shows the average computation times under two scenarios on one computer. As the penetration rate increases, the computation time increases as well. In this traffic monitoring system, an increase in penetration rate does not increase the complexity of the Kalman filter for the same scenario, so the main factor that contributes to a larger computation time is the increase in running SQL query and collecting BSMs from the server. With a larger penetration rate, more BSMs need to be collected and processed. In scenario 1, it took 4927 seconds (1.4 hours) to finish the computation for 10 RSUs. As the duration of the experiment in AIMSUN is set to be 12600 seconds (3.5 hours), the traffic monitoring system can be implemented in a traffic network when the penetration rate is about 20% with only one computer. For this scenario with 20% BSMs, if we use ten computers to run the experiment, each computer will only spend 9 minutes on the traffic state estimation. Moreover, we find all experiments for scenario 1 can be run in real-time using

only one computer as the computation time for 10 RSUs is smaller than the simulation time (12600 seconds (3.5 hours)). In scenario 2, the system cannot run in real-time with only one computer. All experiments for scenario 2 have a computation time longer than the simulation time (12600 seconds (3.5 hours)). The computation time for scenario 2 is longer than scenario 1 because the traffic volume and the detection area in scenario 2 are much larger than those in scenario 1 and the Kalman filter in scenario 2 is more complicated than that in scenario 1, which has more cells and inter-cell interaction. For scenario 2, multiple computers can be used to compute the traffic state and to make the system work in real-time. When the penetration rate is 20%, there should be at least two computers to make the system operate in real time. When the penetration rate is 80%, the computation time is 36237 seconds (10 hours), which is one hour for each RSU on average. A computation time of one hour for each RSU is still smaller than the simulation time of the test. It means if there 10 computers running programs for 10 RSUs separately, the system can still operate in real time.

Table 13: Computation times

scenarios	penetration rate	computation time	% of real time
10RSU	20%	4927 sec/10 RSUs	39%
10RSU	40%	6806 sec/10 RSUs	54%
10RSU	60%	10740 sec/10 RSUs	85%
10RSU	80%	12087 sec/10 RSUs	96%
I35	20%	13213 sec/10 RSUs	105%
I35	40%	19468 sec/10 RSUs	155%
I35	60%	25419 sec/10 RSUs	202%
I35	80%	36237 sec/10 RSUs	288%

10.2.5 Estimation accuracy

The estimation accuracy of the system is evaluated with the ground truth traffic states. Figure 104 shows the estimated density as well as the actual density measured by using BSMs from 100% equipped vehicles in scenario 1. Each row of the heatmap corresponds to a cell. The first to the last cell corresponds to each cell in the detection range of each RSU listed in Table 11. As the first RSU in Table 11 is RSU 10 and this RSU has three cells, so the first three rows in the heatmap show the result of RSU ”rsu_62_WB”. The detection area of an RSU can be divided into multiple cells.

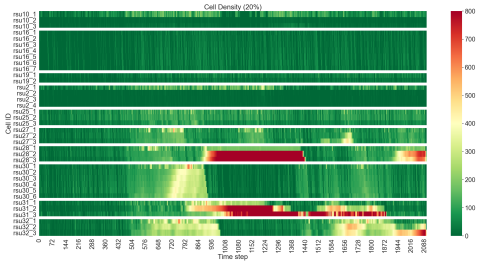
In the last three rows of the heatmap of Figure 104e, there are two time periods that the road is congested. All estimated results show these two congested periods. However, for the fourth to

the last RSU (RSU 28), the error is large. This section is estimated to be congested, which differs from the actual condition. The estimated result for all uncongested road sections seems to be reasonable. When the penetration rate is more than 60%, the estimated result show to have two congested time periods for RSU 28.

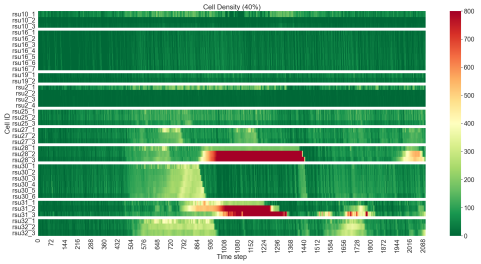
Figure 105 shows the speed estimation in scenario 1. Figures 105c and 105d are more similar to the actual speed measurement. When only BSMs from equipped vehicles are collected, as these vehicles are not uniformly distributed in the network, their travel states will represent the traffic condition of the area they are currently in. In Figure 105b, the right bottom corner of the heatmap does not show a heavy traffic due to a normal average speed of detected vehicles. Figure 106 shows the flow estimation. All estimations show an similar pattern as in the heatmap for actual flows except for RSU 31.

Figure 107 shows the density estimation in scenario 2. When the penetration rate is lower than 80%, the traffic congestion only generates in the detection area of RSU 43 but the congestion should propagate to the detection area of RSU 65. The estimation result with 80% penetration rate can produce this phenomenon. In the speed estimation (shown in Figure 108), we can find that the estimation result overestimates the congestion generated at the upstream of freeway, which is in the detection areas of RSU 68 and RSU 37, but underestimates the congestion at the downstream of the freeway. Figure 109 shows the flow estimation in scenario 2. The flow estimation is calculated based on the density estimation, so the heatmap of flow estimation has a similar pattern to the heatmap to the density estimation. However, the actual flow shows a different pattern from the actual density when comparing Figure 109e and Figure 107e).

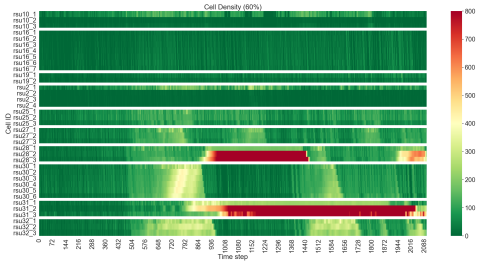
Tables 14, 15, 16, and 17 show the estimation errors under different penetration rates for two scenarios. As penetration rate increases, there is no clear trend showing an increase in the accuracy. As the penetration rate is set for the entire network rather than for the target area so the actual penetration rate for each area may differ. A increase in the penetration rate for the entire network may not guarantee an increase in the penetration rate in the target area. Besides, in the experiment, the entering flow to the section and the flows on on ramps are unknown, which contributes most errors in the system. In this system, we estimated these input flows using the measured flow divided by the penetration rate. As the penetration rate set for the entire network is not equal to the penetration rate on the target road section, increasing the penetration rate does not result in a better estimate for all input flows.



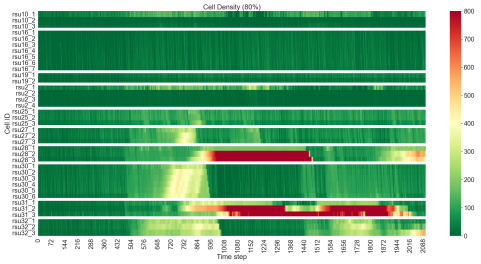
(a) 20% penetration



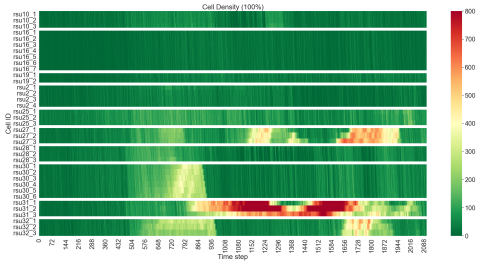
(b) 40% penetration



(c) 60% penetration

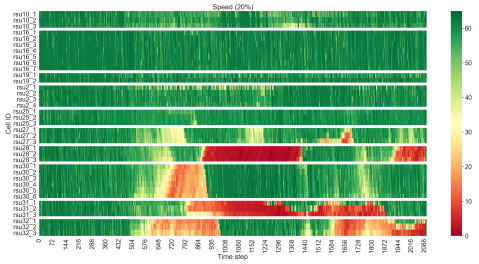


(d) 80% penetration

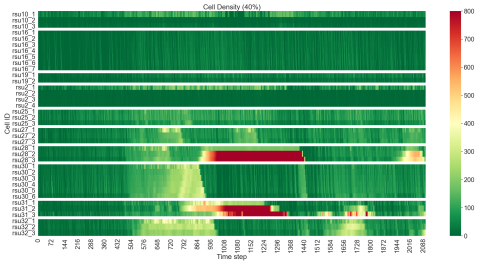


(e) 100% penetration

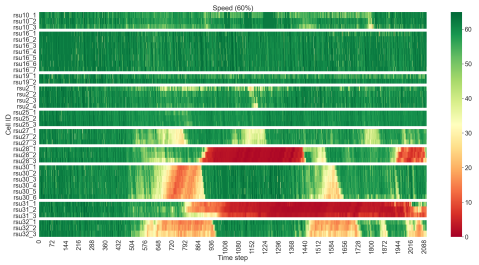
Figure. 104: Estimated densities in scenario 1



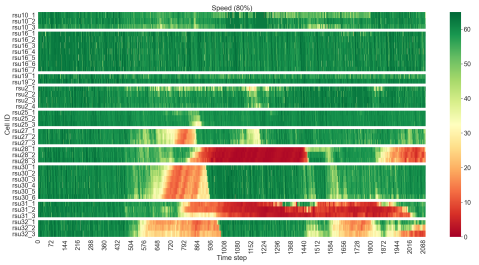
(a) 20% penetration



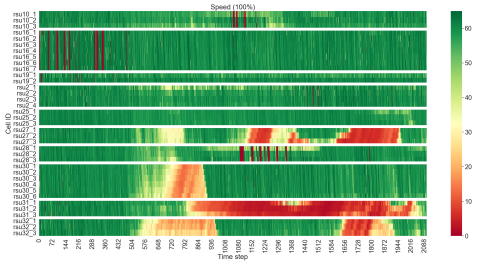
(b) 40% penetration



(c) 60% penetration

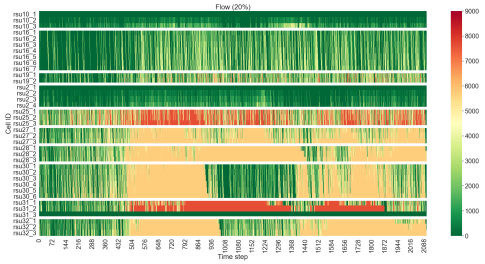


(d) 80% penetration



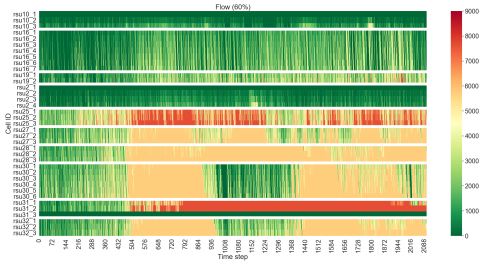
(e) 100% penetration

Figure. 105: Estimated speeds in scenario 1



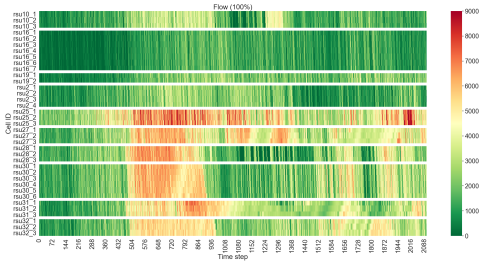
(a) 20% penetration

(b) 40% penetration



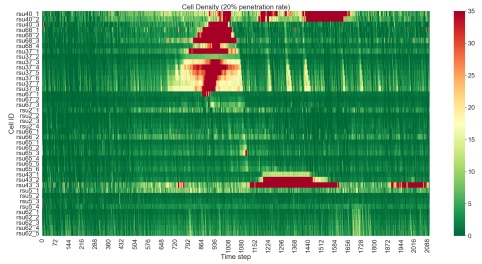
(c) 60% penetration

(d) 80% penetration

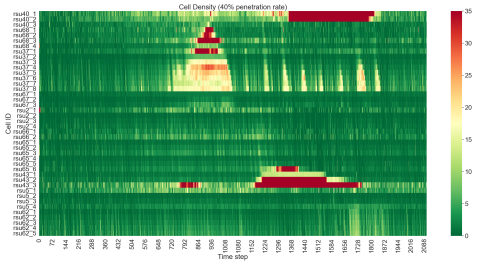


(e) 100% penetration

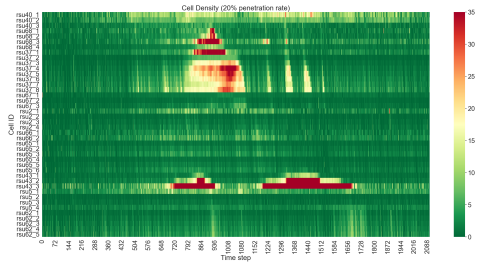
Figure. 106: Estimated flows in scenario 1



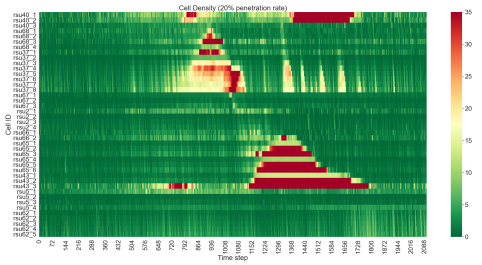
(a) 20% penetration



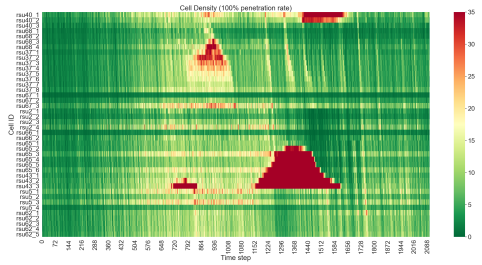
(b) 40% penetration



(c) 60% penetration

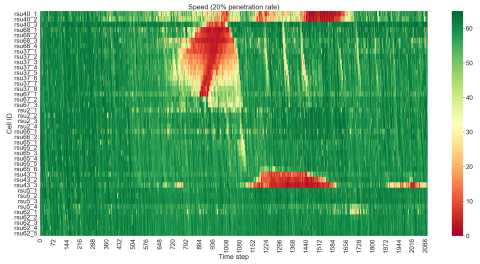


(d) 80% penetration

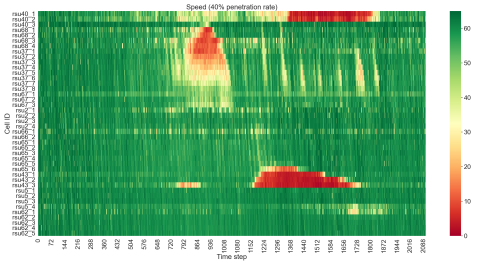


(e) 100% penetration

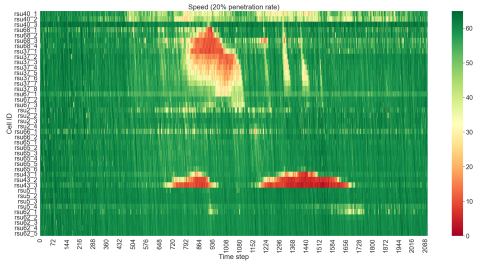
Figure. 107: Estimated densities in scenario 2



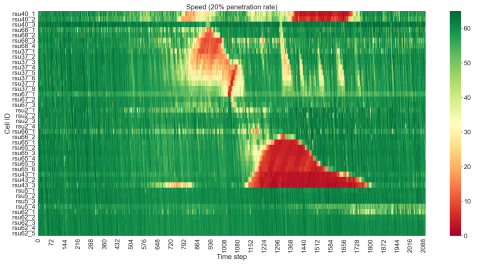
(a) 20% penetration



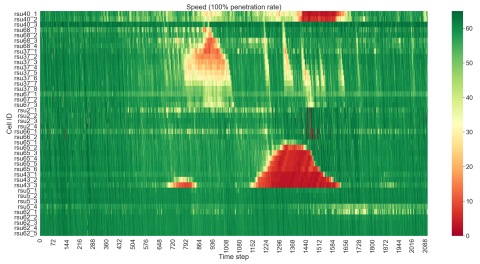
(b) 40% penetration



(c) 60% penetration

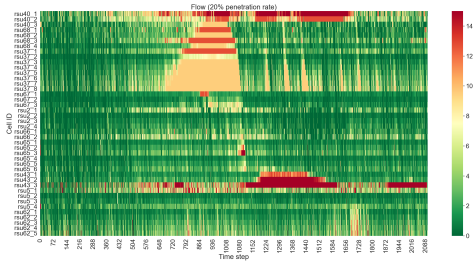


(d) 80% penetration

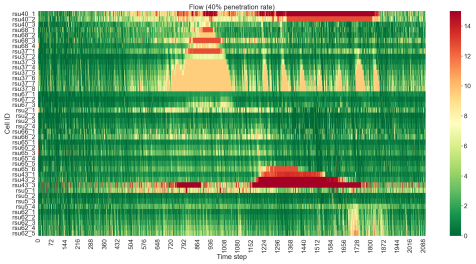


(e) 100% penetration

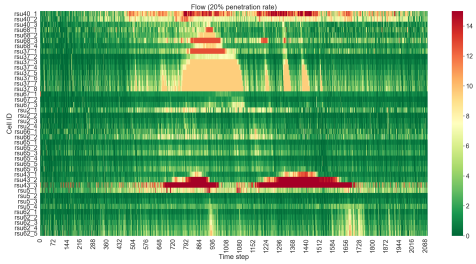
Figure. 108: Estimated speeds in scenario 2



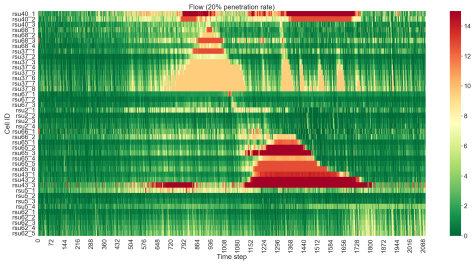
(a) 20% penetration



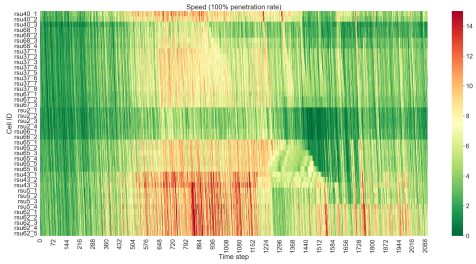
(b) 40% penetration



(c) 60% penetration



(d) 80% penetration



(e) 100% penetration

Figure. 109: Estimated flows in scenario 2

Table 14: Estimation error for scenario 1 (absolute value)

scenario	penetration	density error	speed error	flow error
10RSU	20%	96.5 veh/mile	7.52 mile/hour	2586 veh/hour
10RSU	40%	98.2 veh/mile	7.67 mile/hour	2586 veh/hour
10RSU	60%	90.7 veh/mile	7.13 mile/hour	2586 veh/hour
10RSU	80%	93.9 veh/mile	7.07 mile/hour	2580 veh/hour

Table 15: Estimation error for scenario 1 (relative error)

scenario	penetration	density error	speed error	flow error
10RSU	20%	0.90	0.14	0.99
10RSU	40%	0.91	0.14	0.99
10RSU	60%	0.84	0.13	0.99
10RSU	80%	0.87	0.13	0.99

Table 16: Estimation error for scenario 2 (absolute value)

scenario	penetration	density error	speed error	flow error
I35	20%	82 veh/mile	6.17 mile/hour	3498 veh/hour
I35	40%	83 veh/mile	5.16 mile/hour	3174 veh/hour
I35	60%	84.6 veh/mile	5.23 mile/hour	3174 veh/hour
I35	80%	79.6 veh/mile	4.74 mile/hour	3174 veh/hour

Table 17: Estimation error for scenario 2 (relative error)

scenario	penetration	density error	speed error	flow error
I35	20%	0.91	0.11	0.99
I35	40%	0.92	0.09	0.99
I35	60%	0.93	0.09	0.99
I35	80%	0.88	0.08	0.99

10.3 Discussion

In this section, the project team created a traffic monitoring system to estimate traffic states in a large scale traffic network. In the traffic monitoring system, the data input is BSMs generated by a microscopic simulation model. The data is stored on a server and can be accessed by any granted computer remotely. Scenarios under different penetration rates are used to test the efficiency of the system. The result shows that one computer is enough for running the system in real-time for the first scenario but is not enough for the second scenario. Among all experiments, the one using 80% penetration rate in scenario 2 takes the longest computation time (10 hours) on one computer, but if there are multiple computers running programs for each RSU separately, the system can operate in real time. Computation time may further decrease in the future due to improvements in computer processor technology and/or further improvements in the traffic state estimation algorithm. If traffic state estimation is performed for each RSU individually, one computer could process around 6–7 RSUs. Based on the number of RSUs required for the entire freeway network, these numbers may provide an estimate for the computational resources required.

11 Research Benefits and Implementation Steps

BSMs include vehicle travel information such as the coordinate, speed, and acceleration, which can be utilized to estimate traffic states on the road. However, since the regulation requiring vehicles to have this functionality has not yet been made official, it may take several years before a significant number of vehicles will even begin to transmit this information. Still, the proper methodology and models can be developed now to ensure that the state can immediately take advantage of BSM technology when it is inevitably implemented into vehicles. Even a small market penetration of BSM-transmitting vehicles provides traffic state information through their speed choices.

11.1 Benefit

The expected benefits from this research encompass several categories. Table 18 below provides an overview of these benefits.

11.1.1 Improved Traffic Information

Traditional loop detectors are installed at several spots on the road while BSMs can cover the entire road section. Loop detectors can only provide the traffic state measurement for the position where they are installed. The traffic states include flow, time-mean speed, and density. Traffic density is not the direct measurement of loop detectors but is calculated with measured flow and speed. The measured time-mean speed is the average speed of vehicles crossing the loop detector and it cannot reflect the traffic condition for a road section. The traffic state estimator using BSMs in this study provides traffic state estimation for each cell of the road section, including traffic density, flow, and space-mean speed. Space-mean speed is the average speed of all vehicles that pass through a target area on the road, which can be a better representative of traffic efficiency.

Table 18: Expected Research Benefits

Benefit Category	Key categories applicable to this project	Are the benefits quantifiable (Yes/No)	How will these benefits be quantified?
Improved Traffic Information	X	Y	Before/after traffic information accuracy comparison (e.g. accuracy of travel time)
Decrease Engineering/ Administrative Costs Reduce Road User Costs Reduce Environmental Impact Improved Safety	X	Y	Comparison of travel times
Operation and Maintenance Saving Reduce Risk	X	Y	Monetary difference between fixed sensors versus BMS equipment

The test results all show that the speed measurement has a high accuracy even with small market penetration. When the traffic state for the entire road is given, the road operators can update their current strategies to take advantage of the higher resolution of the space and time traffic information inferred through the BSMs. It can also help road operators to upgrade their road control policies to new ones that require richer input data.

11.1.2 Reduced Road User Costs

Travel times experienced by drivers in the state will also see improvements. When improved traffic information are used in traffic control, such as ramp meters and adaptive signals, the traffic efficiency can be improved because of more precise utilization of travel controls. Previous studies have shown that adaptive signal control can improve travel time by 10 percent or more (28). Additionally, a MnDOT study of ramp metering in the Twin Cities area noted 22 percent in savings for freeway travel time (38). The benefits of these signals and ramp meters can be better reaped if more accurate traffic information is available. Besides, if drivers can get access to the traffic state estimations, they can switch to a path with a higher travel speed than the current path, which may reduce the travel time for their trip.

11.1.3 Operation and Maintenance Savin

As the penetration rate of BSM-equipped vehicles increases over time, less reliance on fixed sensors will be needed. Loop detectors at one end of the road section may still be needed to improve the estimation accuracy. As the amount of the loop detectors reduces, the need and cost to maintain the replace these fixed sensors also decrease. The installation costs for one loop detector sensor can be somewhere around the $500 range, and with the temperamental quality of these sensors, the replacement of them can end up being semi-frequent (29). This will lead to both monetary savings when considering the cost of purchasing the sensors, and savings in labor because there will be less needs to maintain or repair these sensors.

11.1.4 Implementation

At first, a traffic state estimation algorithm was developed based on the Kalman filter technique. The Kalman filter technique is used to integrate the prediction model and measurement. In this project, the cell transmission model is used as the prediction model to predict the traffic state. The measurement in this project is speed measurement extracted from BSMs. The Kalman filter is able to generate an estimation with a smaller uncertainty than both the predicted value and the measurement. The traffic state estimator is able to estimate the traffic state under different

penetration rates, as well as integrating data inputs from BSMs and loop detectors. When the cell transmission model is used as the traffic flow model, the target road section is divided into several short road segments called ”cell”, which is the basic unit for the traffic state estimation in this project. The output of the traffic state estimator is the density, speed, and exiting flow for each cell. This part of the work is covered by Task 1 and Task 2. After the development of the traffic state estimator, the accuracy of the traffic state estimation was evaluated with different test datasets, including BSM datasets generated from a CTM-based model, a microscopic simulation model, and radar data. The ground truth databases including the actual traffic state for each cell is also prepared for the accuracy test. To explore the required computing power to run the traffic state estimator in real-time, a traffic monitoring system is constructed, which is consisted of a data collection module, a data transmission module, and a data processing module. The data collection module includes an online server with multiple databases in which the BSMs are generated from a microscopic simulation model corresponding to a large-scale network. This part of the work is covered in Task 3, 4, 5, and 6. If this traffic state estimator is implemented in the field, some infrastructures are required. First, there must exist some type of infrastructure which is capable of serving as receptors for the BSM information that is being broadcast. These receptors will need to be placed at specific points where vehicle data is of interest. A study from NHTSA entitled “Vehicle-to-Vehicle Communications: Readiness of V2V Technology for Application” details information about cost estimates of implementing equipment for BSM technology (15). In this study, it is estimated that a DSRC roadside receptor will initially cost approximately $8,800 per site. This is based on the assumption that the receptor would need to be replaced once every fifteen years and that there is an annual maintenance cost of approximately $7,500 per site. To put it into perspective, the study estimates that 19,750 roadside receptors will be needed to cover 74% of the nation’s population. See Table 19 blow from this study which summarizes the costs of DSRC receptors.

12 Functional Specifications

12.1 Introduction

Connected Vehicles (CVs) have the potential to change transportation fundamentally. Though CV applications for safety, mobility and environmental benefits are exciting, the implications of BSMs that all CVs will broadcast must also be considered. Can a BSM-based monitoring system (BSMMS) potentially replace current traffic measurement collection infrastructure? The following is a consideration of functional specifications for a BSM-based system to be able to be used for gathering traffic metrics, based on literature. For the purposes of this report, DSRC is assumed to be the most common way of broadcasting and receiving BSMs. Other technologies such as Auto5G and Cellular-V2X (C-V2X) are being considered by certain companies for their existing infrastructure (cell towers) and range (wherever there is cell service). A drawback of C-V2X is potential cost as well as turning traffic data over to third party companies. Again, this report assumes DSRC will be the primary BSMMS and elaborates how and if the cellular communication path makes a difference.

12.1.1 Current Traffic Data Collection System and needs

In order to better define the functional specifications of a BSMMS it is important to summarize the currently used traffic data collection systems and more importantly the critical systems they support. It stands to reason to assume that moving forward with a BSMMS means expanding traffic operation and traffic monitoring abilities without compromising current critical operations.

The Minnesota Department of Transportation (MnDOT) Regional Transportation Management Center (RTMC) oversees the systems controlling close to a million vehicles daily in the Twin Cities area. The RTMC's goals include:

- Reduce crashes and congestion
- Increase freeway capacity
- Increase freeway speeds during peak periods
- Provide accurate, timely traffic information to travelers
- Remove stalled vehicles by FIRST
- Assist State Patrol with Incident management
- Enhance service for transit and carpoolers

To support these goals, the RTMC requires data from the roadway, both real time and historic. To increase safety and throughput on existing roads, there are two options: reconstruction, or operating the road more efficiently. Reconstruction involves looking at historic, sampled, and discontinuous data. This data could be aggregated; fine, real time data is not required to make decisions about road geometry. In many cases, the expense and time of reconstruction make it a suboptimal solution. Many of the RTMC's systems and functions instead focus on operating the existing roadways more efficiently.

12.1.2 How are traffic metrics captured today?

RTMC equipment includes, as of 2014:

- CCTV Freeway Cameras – 600
- Loop Detectors in roadways – 5500
- Radar Detectors (Wavetronix) – 230
- Lane Control Signals – 300
- Dynamic Message Signs – 184
- Ramp Meters – 443
- HOT lanes
- 511 information

The loop detectors are single detectors; they can only provide an estimate of average speed to be used to determine density, along with flow, which is derived from counting the number of vehicles passing the detector in a 30 second period. MnDOT has also begun to deploy Wavetronix radar, which captures true speed rather than an estimate of it. Even with accurate speed, density remains an estimate.

Reducing Crashes and Congestion

Crash data is collected to identify dangerous areas of the roadway. To discover which areas of the roadway are experiencing abnormal amounts of crashes that need to be addressed, the number of crashes per volume of the roadway needs to be considered. The Average Daily Traffic (ADT) on the roadway is required to determine where an abnormal amount of crashes is happening.

To measure congestion on the roadway, a benchmark speed must be set and measured. MnDOT considers anything less than 45mph on the freeway to be congested; how much congestion a freeway experiences is measured by how many 30 second intervals the traffic moves at less than 45mph per day.

Another aspect of congestion is how long it spreads on the roadway. Loop detectors are placed every half a mile in an effort to capture gapless data from the roadway. This allows congestion to be tracked; the more seamless the data, the more accurately the cause and extent of congestion can be identified.

When deciding if reconstruction is appropriate for a roadway, a benefit cost analysis must be conducted with available data. If the number of crashes, hours of congestion or number of people affected by it are not enough, it may not be worth reconstructing the roadway. The total delay caused by congestion must be calculated. This is done using speed and the number of vehicles travelling at that speed; the delay of the average vehicle multiplied by the total volume on the roadway gives the total delay, or the amount of money lost to congestion.

Improving Freeway Operations

To improve operations, real time data is required; speed and volume dictate the quality of the traffic and the driver experience. Capturing the finest data possible will allow for better decisions to be made, and better strategies and policies to be implemented to affect the current roadway without the time or expense of reconstruction.

The RTMC's goals of Increasing freeway capacity and speeds during peak periods are both another way to say increasing throughput. The freeway capacity is controlled by the environment and geometry; the only way to directly change capacity is by widening a facility. To increase throughput, the facility must be operated as close to capacity without congestion; this will produce throughput and increase the production of the roadway.

So if the geometry cannot be changed to easily accommodate higher numbers of vehicles, how can the throughput of the roadway increase? The RTMC offers many services and systems to reduce congestion. If congestion is lessened and roadway speeds increased, the throughput will increase as well. Additionally, systems like ramp metering and driver information (either on the roadway, or through services like 511), help keep the volume on the roadway within an acceptable level. Controlling the number of vehicles on the roadway is another way to increase capacity for vehicles on that roadway.

Controlling Demand

Three strategies and systems that the RTMC uses to control the number of vehicles on the general purpose roadway include ramp metering, MnPASS (also known as HOT lanes), and providing information to drivers to influence their decision making.

The first ramp meter in MN was installed in 1969 along I-35E; in 2019, there are 433 ramp meters active in the Twin Cities metro area. Ramp metering helps reduce congestion on the freeway as well as back up from the freeway entrance ramp. Ramp metering looks at the average density of all general purpose lanes, but uses some lane-by-lane data to ignore traffic in auxiliary lanes or MnPASS lanes. In 2001, when ramp meters were turned off for a period of time, a significant spike in congestion on the freeways was observed. Ramp metering depends on density estimates (based on volume and density) reported up to three miles away from the ramp in real-time; ramp meters are adjusted every 30 seconds based on that data. When studied in 2000, ramp metering resulted in over 25,000 hours of travel time saved, much more reliable travel time than without, safer roadways, and a net monetary benefit outweighing the system's cost to operate. Though drivers may not appreciate ramp metering while they are queueing on the ramp, the system certainly is beneficial to keeping highways safe and as uncongested as possible, and depends on accurate and timely measurements from the rest of the roadway to function.

MnPASS, MnDOT'S High Occupancy Toll (HOT) lanes, relies on density estimates to function properly. MnPASS lanes are priced dynamically based on density in the MnPASS lane only. Detectors in the pavement measure the traffic volume in 30 second intervals and gather average speeds to estimate the density in the lane; as congestion in the lane increases, the fee to drive in the lane increases, and as it decreases, so does the fee. Note that this only looks at the MnPASS lanes, not general purpose lanes. By federal law, speed in the MnPASS Express Lane must be at or above 45 miles per hour for 90% of the time; currently, speeds are at 45 miles per hour approximately 95% of the time, and MnDOT reports that during peak-travel times, one MnPASS lane can carry the same number of people as two general purpose lanes. Overhead digital (VMS) signs show the current price for one or two segments along the corridor. The price on the sign is updated every three minutes depending on the current demand in the MnPASS lane.

The public makes use of real time data provided by the RTMC through the 511 phone line, website, and app. 511 provides users with congestion, construction, and incident data. In 2014, almost 300,000 calls to 511 were reported. The website had nearly 4 million visitors, and the app had 92,000 downloads in less than a year from its launch. 511 provides varying levels of detailed traffic and incident reports, depending on which view a user chooses. Additionally, 511 offers support for the trucking industry, with a trucking-specific section of the website. Services

like Google Maps also take data from the 511 system to supplement the probe data from mobile phones and to alert drivers to incidents or congestion. With all this information readily available at near real time, drivers can make educated choices about which path they would like to take to their destination.

Improving the Freeway for Remaining Vehicles

There is a fine line between a freeway operating near capacity and one that has broken down. A system functioning well will still have smooth traffic even with high demand, ideally with drivers making no sudden stops to lead to shockwaves or crashes. Real time tools such as:

- Travel Time Information Displays
- Advisory Speed Limits
- Queue Warning
- the Smart Work Zone Speed Notification System (SWZN)
- Variable Message Signs (VMS) or Dynamic Message Signs (DMS)

Can all help influence drivers on the roadway to anticipate congestion and react appropriately as they approach it. Most of these applications depend on real-time, lane-by-lane data. The travel time information algorithm, like ramp metering, ignores speeds in both auxiliary lanes and MnPASS lanes. It focuses on the general purpose lanes' speeds to calculate the travel time, and display it to drivers. The RTMC does not post travel time information in the MnPASS lane, but it does look at travel times there to monitor performance.

Though advisory speed limits are no longer used in Minnesota, when it was operational, the system looked at average speeds in the general purpose lanes to display warnings to drivers. The system didn't post lane by lane advisory speed limits, but needed lane by lane data to ignore auxiliary lanes and MnPASS lanes.

Queue warning, though still experimental, looks at average speeds in the general purpose lanes to warn drivers of upcoming congestion. MnDOT has used a type of Queue Warning, the Smart Work-Zone speed Notification (SWZN) system, to specifically notify drivers about changing conditions on the roadway as they approach a work zone.

Currently, lane control signals (LCS) are operated in Minnesota by trained camera operators, not automatically based on an algorithm. Queue warning strives to be fully automatic; detect the queue, turn on the driver warning sign. Additionally, DMS may be used to reroute vehicles from incidents.

Incident Management

To manage incidents, MnDOT uses a fleet of FIRST patrol trucks, and assists the State Patrol with diverting traffic. Not all of MnDOT's functions require traffic metrics to operate. For example, FIRST responders detected 85% of incidents by driving their assigned routes and noting vehicles causing issues with traffic flow, rather than from RTMC alerting them to an issue. If a detector has a sudden spike in congestion and an incident is suspected, the call will still go out to FIRST from the RTMC. Before drivers began carrying cell phones, this was a much larger part of FIRST response.

A part of assisting the State Patrol with incident management is making sure that traffic diversions due to incidents are appropriate for the volume on the roadway to be diverted. The ADT of the roadway and capacity of other roads for diversion must be known. Historic data can be used for determining demand, by looking at finely spaced measurements throughout the day of volume and space on roads; real time data can be used to look upstream, see the volume that is approaching the incident site, and find an appropriate roadway to divert it to. Integrated Corridor Management may also be used when diverting traffic onto a signalized road, to change signal timing based on the anticipated volume from the diverted road. The best way to appropriately divert traffic would be to monitor the entire corridor– freeway and arterials– for real time volume.

Transit Enhancement

Transit and carpools are able to use the MnPASS lane for free. This encourages less overall vehicles on the roadway, due to the higher capacity of transit vehicles and the requirement of at least two passengers to use the MnPASS lane freely during peak hours.

Tools for Real Time Data Capture

Currently, the RTMC uses CCTVs for operation and traffic control, DMS for driver communication, and LCS for direct effects on the roadway. Loop detectors and radar are used for information gathering, both historic and real time.

12.1.3 What do connected vehicles broadcast?

A BSM is a packet of data broadcasted by a Connected Vehicle (CV). BSMs can be received by other vehicles (V2V), by infrastructure (V2I), or by anything else that is listening (V2X). BSMs are used for various applications, falling under the headings of safety, mobility and environmental (6). BSM format is designed for radio frequency economy and low latency, aimed primarily to localized broadcast required by V2V safety applications. Core information (group 1, below) about

Table 19: Types of information transmitted by BSMs

Group 1 (transmitted at 10hz)	Group 2 (transmitted as needed by applications or road events)
Timestamp	Recent braking
Position (latitude, longitude, elevation)	Path prediction and history
Speed and heading	Throttle position
Acceleration	Additional information about vehicle type
Brake system status	Automated Braking System (ABS)
Vehicle size	Stability Control
Steering Wheel Angle	GPS status and data
Positional Accuracy	Weather predictors (lights, wipers, traction, etc.)

the vehicle is transmitted 10 times a second. Supplementary information (group 2, below) can be transmitted at a slower rate, as needed dependent on road events such as automated braking system activation or weather applications. The data packet will vary from CV to CV, but generally includes the following:

Data regarding fuel usage, road grade, engine drive cycle and temperature, and operating mode are not planned for BSM usage. Additional environmental information can help provide weather warnings for other drivers or eco-trajectory computations, but is not required.

12.2 Functional specifications

To understand how a BSMMS can be developed and eventually implemented, this report examines simulations, real-world test pilot sites, and other literature available. Lane accurate data is discussed first, as it is necessary for all other functions of the BSMMS. In addition, data security and driver privacy efforts are discussed, as gathering travel time estimates may impact them.

12.3 Lane accurate data

Historically, basic traffic metrics were captured by single- or double-loop detectors, while lately newer technologies like video and radar are increasingly used to reduce cost and increase accuracy. By virtue of the loop detectors being buried in specific lanes, lane accuracy of gathered information is essentially assured. Regardless of the actual technology implemented, existing measurement systems are location based, fundamentally incapable of measuring lane Density or Space Mean Speed (SMS), the two metrics required along with Flow to describe any traffic condition. At best case scenario, double loops, video, and radar detection systems in addition to lane counts (Flow), can directly measure Spot Speed (Arithmetic mean Speed). These two measurements, combined

with a fair amount of assumptions and traffic flow modeling are used to estimate, with a fair amount of error, either SMS or Density (14). So, although lane specific, accurate measurement of flow and fair estimates of density are available at fixed locations. Gathering traffic information between locations takes an additional level of estimation and results in further loss of accuracy.

The fundamental paradigm change of the BSMMS is that, at 100% market penetration, it can directly provide measurements of Density, Flow, and potentially SMS.

BSM GPS Data Accuracy

The most fundamental component of a Connected Vehicle is the onboard GPS. Although other assistive systems can exist in the vehicle, the location included in the BSM originates from the OBU GPS. In addition, when the vehicle is near an RSU, it will receive broadcasts with GPS accuracy corrections. Therefore, the aforementioned need for lane accurate information depends on the ability of the OBU GPS to achieve sub-lane accuracy.

A 2017 BSMMS study by ImageSensing Systems (37) was performed using vehicles equipped with OBUs traveling at a short stretch of road equipped with RSUs, while infrastructure sensors used to generate emulated BSMs. The goal of the project was to utilize new kinds of sensors in order to produce estimated BSMs for non-instrumented vehicles, but since ground truth was a basic need in the research project, an evaluation of the OBU GPS accuracy was also performed.

Figure 110 shows the contrast between externally gathered vehicle information and OBU GPS data. Based on the error, GPS data alone would not provide accurate lane information (or at least as accurate as radar is assumed to be).

As of 2008 the civilian GPS standard for accuracy on a moving target is roughly 4 meters. This can be affected by weather, atmospheric conditions, or other issues affecting communication with the satellite system. To account for this, researchers will need to develop algorithms to correctly project the BSM data on to the roadways, and ensure the vehicle information is shown in the correct lane. In addition, high resolution maps will be required to project the data on the road network for use in the BSMMS.

BSMMS Accuracy

The idea that BSMs can replace current traffic detector systems is ambitious; it has been touted as capable of providing real-time (4), accurate traffic detection in addition to all of the applications for safety, mobility and the environment (8). A CV pilot site in Tampa has been operational since fall of 2018, and researchers are posting data gathered by BSMs transmitted to infrastructure sensors and Road Side Units. Tampa's dataset was chosen over the other two test sites in New York and

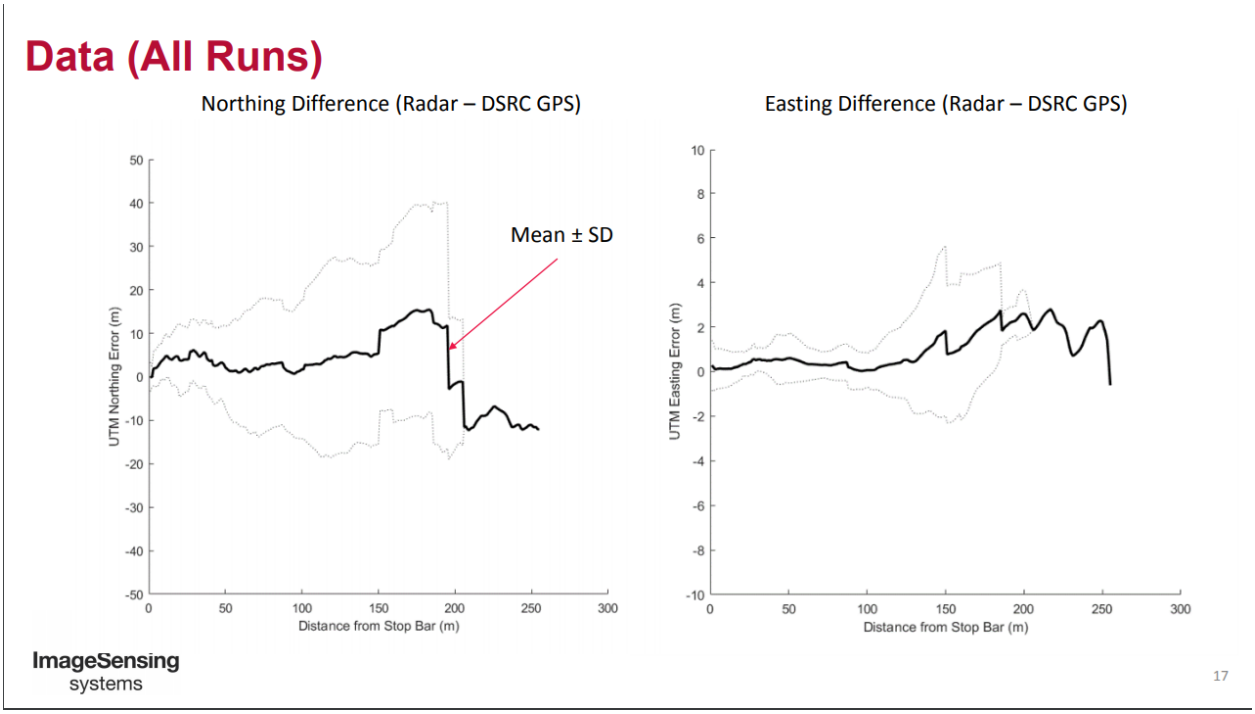


Figure. 110: Slide showing the error between traditional radar detected northing and easting, and OBU GPS data

Wyoming because it was the largest and most accessible at the time of writing. Figure **??** below is an example of a speed map generated by the BSMs from the connected vehicle fleet that Tampa has deployed. Note, speeds are expressed in 0.02m/s. A value of 100 on the chart is equivalent to 2 m/s (7).

This system is currently not producing real-time maps of the CV site. But can the BSMMS provide location-accurate historical data?

As shown in Figure 112, on closer inspection of the data points, many vehicles are somewhere on the roadway. However, it is clear that GPS inaccuracy or other issues transmitting and receiving BSMs has caused many vehicles to be slightly off the road or even off it entirely.

While lane by lane data is not a requirement for some traffic metrics or future applications, it is provided by the current loop detector instrumentation. To switch from that system without compromising the utility of information received through BSMMS, it must continue to provide lane by lane data.(Lane by lane requirements per application, and approach that requirement in a different way—ramp metering doesn't require lane by lane, but it does require the exclusion of auxiliary lanes with non-homogenous traffic patterns, so they don't affect the general estimate of density—potentially excluding by filtering out outliers instead of exclusively lane by lane as is done now)

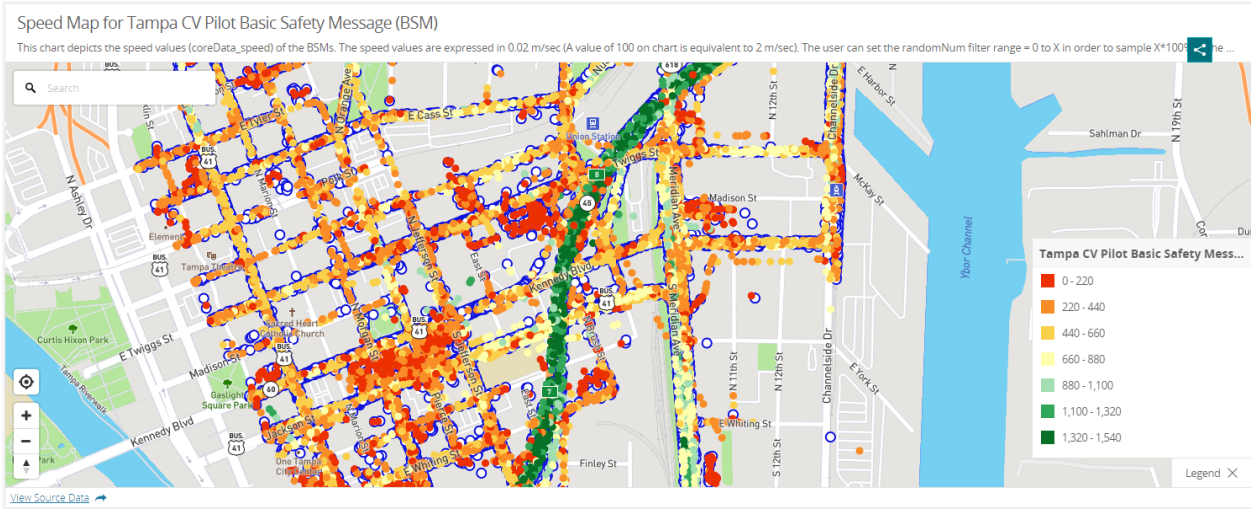


Figure. 111: Wide view of Tampa CV data, taken March 20th 2019

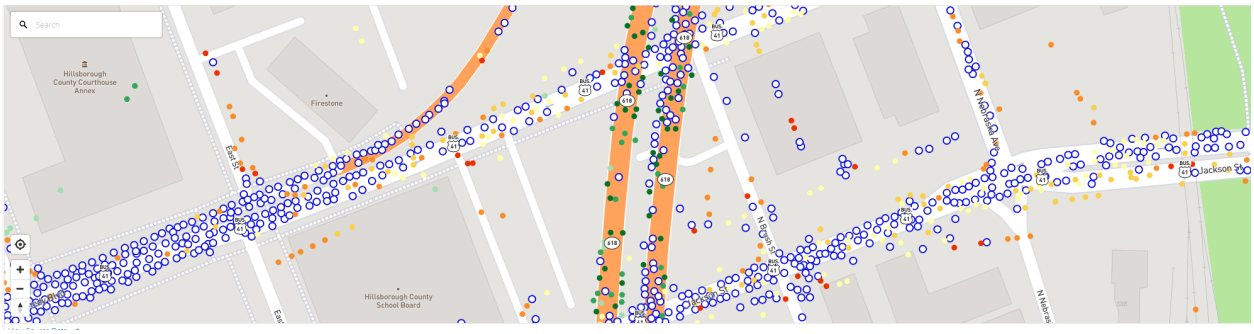


Figure. 112: Close up of Tampa CV speed map

SPECIFICATION 1

BSMMS must provide lane by lane data.

- BSMMS need to be able to take the information provided by the BSMs and determine which roadway and which lane of that roadway each vehicle is traveling.
  - As explained, currently the GPS alone is not able to reliably provide location accurate enough to assign the vehicle to a lane. Therefore additional post-processing algorithms are necessary to refine the raw information provided by the BSMs. For example, if one has accurate, up to date high resolution maps, containing the actual lane boundaries, with accurate lane widths, especially on sections of road where there are transitions, it is conceivable that an algorithm can be developed which based on relative lateral differences between vehicles driving on different lanes as well as their heading and speed, relative lanes can be assigned to each vehicle (rightmost, second

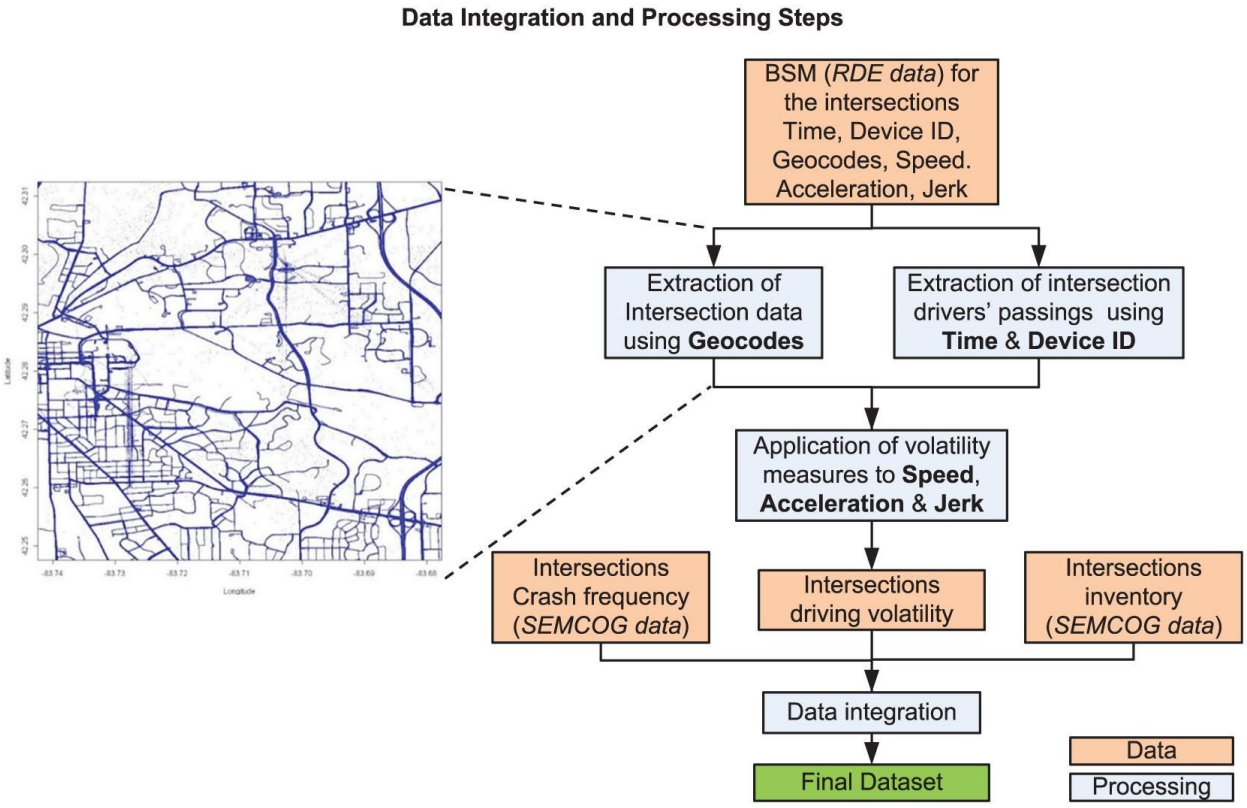


Figure. 113: Map generated with two-month BSM geocode data from Ann Arbor, MI

from the right, etc.) and be compared with the actual geometry of the road as described in the digital maps. The result will be a better estimation regarding the lane each vehicle is.

12.3.1 Speed

Currently, loop detectors and radar are used to gather vehicle speed information. Excluding some isolated cases like the I-94 CV Testbed utilized in this project, current sensors provide only spot-speed measurements. Making the assumption that this Time-Mean-Speed (TMS) is reasonably close to Space-Mean-Speed (SMS), one can use it to estimate the density of a given roadway segment. Traditionally, speeds are gathered every half a mile on freeway systems.

BSMMS have the ability to gather space-mean-speed in addition to spot speeds, as vehicle speed is transmitted along the entire length of the detector range rather than just a point. Presumably, speed information will be spot captured along the length of the detector, but algorithms could be developed that would knit together the BSMs to create space-mean-speed. Capturing these speeds would be one method of deriving travel time and end of queue locations.

SPECIFICATION 2

BSMMS must gather speed information as it is transmitted from vehicles.

- Space-mean-speed can be calculated with an algorithm connecting the 10 Hz frequency transmissions along the area of radio coverage of one or more RSUs.

12.3.2 Vehicle Counts

Historically, one-off counting efforts have been employed in traffic metrics to gather vehicle count. BSMs offer the possibility of having access to a vehicle count at any point in the stored BSMMS data. So far, no studies have been done solely on the accuracy of real-world BSMs vehicle counts compared to traditional infrastructure capture. Although, accurate counts and, to the extent currently possible, calculation of traffic flow is the most accurate traffic measurement currently available through traditional location based sensors, the same functionality depends greatly on the CAV market penetration. Basically, although a relatively small number of sampled vehicles can produce a fairly accurate estimation of speed, if there is not a complete market penetration for CAVs– or worse, the fraction of CAVs on a particular roadway segment fluctuates– it is impossible to reach a satisfactory estimate of volume.

12.3.3 Roadway Density

As discussed in the last two sections, SMS is a traffic metric that can be estimated with relative accuracy even if the level of market penetration for CAV is low. Density estimates have the exact same issues as volumes. Basically, both metrics involve a direct vehicle count; volume is a count in time while density is a count in space. Excluding the extreme future case of guaranteed 100% market penetration, obtaining estimates of volume and density will involve either a large number of assumptions regarding the shape of the fundamental diagram on each section where accurate SMS is available, or location based sensors are needed to supplement the BSM data. Specifically, if from the BSMMS a reliable estimate of SMS is secured, a vehicle count on that location obtained through traditional detectors will provide a very accurate estimate of density simply through the application of the Fundamental Equation of Traffic.

$$\text{Density} = \text{Flow}/\text{SMS}$$

Further work in ground-truth will need to be done to ensure vehicle counts and density measurements are accurate. As market penetration of CAVs increases and traffic metric capture changes over to using BSMs for data, algorithms to infer true vehicle counts will need to be developed.

A quality of DSRC that is both a boon and a hindrance is its range. The range is estimated at 300m/1000ft with 360-degree capture ability; however, this is affected by line of site. In the NYC test pilot program, RSUs were needed as close as 76m away from each other to ensure the dense infrastructure would not cause gaps in data collection (2). With increased DSRC placement comes increased cost. Cities must determine the resolution of data required for applications, and use that to prioritize placement of the devices. They should be placed so there are no holes or gaps in the data flow.

SPECIFICATION 3

BSMMS must provide accurate vehicle counts.

- This is not currently feasible and additional post-processing based on traffic flow models is necessary.
  - This project proposed a real-time fundamental diagram calibration method and through that it inferred vehicle counts based only on BSM data. The accuracy produced at low levels of market penetration may not be enough for traffic operations needs like ramp metering.
  - It is possible that different methods that fuse information from the BSMs with strategically located infrastructure sensors that provide only counts can improve the estimates of volume and in extend density. This project did not explore this possibility.

12.3.4 Travel Time And Origin Destination Information

BSMs could also be used to estimate travel time based on space mean speed at a snapshot in time, which would not require the use of individual vehicle IDs. However, in order to gather real-time travel time information, BSMs will potentially need to identify individual vehicle IDs along their path. A secure system will need to be in place to ensure only agencies and individuals with proper credentials are reading the BSMs. In addition, there is a need for vehicles to be credentialed to emit BSMs, to prevent rogue information from entering the system. Driver privacy is a concern when accessing the BSM information.

System Security

The USDOT set aside the 5.9GHz broadcasting spectrum specifically to support CVs in 1999. In addition to a dedicated spectrum, security measures are implemented within OBUs and RSUs en-

suring that only actors with proper credentials can exchange data. IEEE 1609.2 lays out standards for DSRC security of transmission of BSMs.

Agencies operating DSRC based RSUs must have a license for operating this equipment in their geographic location, and must register each RSU deployed, both processes handled by the FCC. OBUs do not have to be registered with the FCC as they are automatically enrolled in the CAV system (3).

The Security Credential Management System (SCMS) employed in a DSRC based CAV ecosystem uses a Public Key Infrastructure (PKI) to ensure messages from vehicles and infrastructure can be trusted. New certificates can be obtained on a regular basis to maintain up-to-date information about any untrustworthy entities in the system. This system still being developed; car companies are shifting away from federal government oversight towards private Certificate Authorities (CA), like Green Hills (5), OmniAir, 7Layers and Danlaw (1). The connection to this system for vehicles can be implemented in one of two ways: a DSRC-only approach, which uses DSRC RSUs to distribute keys and certificates obtained from the CA via a backhaul communication line; or a hybrid approach, which uses a combination of cellular, Wi-Fi, and satellite communication to distribute keys and certificates to vehicles. A notable variant of the hybrid approach, in use by the Wyoming DOT CV Pilot, uses SiriusXM satellite communication to push new certificates to the vehicles, with DSRC RSUs only in spot problem locations (4).

As a requirement for the SCMS that DSRC-based CAV implementations relies on, roadside equipment must be connected to the Internet to facilitate communication between the device and the Certificate Authority (CA) that issues signing certificates. This is required in order to sign messages transmitted by the RSU so that they can be trusted by vehicles. This connection may also be used to allow the RSU to occasionally distribute new keys and certificates to vehicle. While the SCMS requires some amount of communication to ensure certificates are up to date, the periods over which these certificates change are not short enough to require a low-latency, high-bandwidth connection, meaning that communication technologies such as DSL or cellular should meet the requirements.

SPECIFICATION 4

BSMMS must comply with security protocol laid out in IEEE 1609.2.

Driver Privacy

In addition to system security, there is a strong concern for driver privacy to be protected. Vehicle IDs assigned to BSMs are temporary, and change intermittently along the vehicle's route;

they contain no Personally Identifying Information and is theoretically impossible to trace back to a single vehicle. An exception may be when outside data could identify an individual vehicle, such as information from a crash report that could be related to vehicle BSM trajectories at the time and location of the crash.

As a response to consumer concerns, the USDOT has stated that V2V systems "will not permit tracking through space or time of specific owners, drivers, or passengers" (3). They also state that BSMs "cannot be used by law enforcement or private entities to identify a speeding or erratic driver" (3). Information transmitted over the CAV network is confidential, and does not involve recording or exchanging any personally identifying information or vehicle movements (3). These privacy measures are supported by the security of the DSRC-based transmission system.

Both travel time and origin-destination (O-D) metrics run into the issue of driver privacy. Travel time requires the identification of a vehicle between two or more consecutive time periods, while O-D information requires identifying an individual vehicle at different locations in the road network. Basically, in either case the metrics come from directly examining a specific vehicle's path.

Travel Time

Presently, travel time information is estimated from loop detectors, radar, and probe-based data sources through a large number of assumptions that affect accuracy. While information from BSMs can be used to estimate travel time without direct vehicle trajectories, BSMs have the potential for much more accurate travel time, at least along traffic links and time periods where vehicle ID keys stay the same. As long as the vehicle ID in the BSM remains fixed the method is identical to current probe-based approaches with vastly higher number of probes available to draw information from. By sampling small pieces of anonymized BSM and using an algorithm to stitch them together, a general estimate of travel time along the link could be created. This would protect driver privacy, as each individual vehicle within a group of vehicles is used for part of the estimate; no one travel path would be exposed. This probe based approach may have more latency than using the space mean speed to estimate travel time, since the probe's travel time along a link is not known until it gets to the end of the link.

The limitation of this method depends on the frequency at which vehicle IDs change; as of right now, SAE J2945/1, (parameter vCertChangeInterval) says the certificates have to change every 5 minutes. This frequency is based on each individual vehicle so it is random when the threshold will be reached for each vehicle. Regardless, barring very congested flow conditions, 5 minutes is a fairly long time period that can cover miles of freeway sections. Driver privacy on a low-density traffic link may come into play; if a driver is the only vehicle ID on a certain traffic link,

their movements are obvious.

Origin-Destination Information

Currently, single-vehicle origin-destination information is gathered through the use of license plate readers. They are generally deployed in isolated cases from a project specific point of view, as the cost of each reader is high and employing them without a specific purpose in mind would be prohibitively expensive. For research purposes, being able to access origin-destination information through monitoring BSMs would be much more efficacious. With vehicle IDs changing at unpredictable moments along traffic links, tracking specific vehicles for origin-destination information is currently not an option for BSMMS. For the purposes of this document, no known solution for using BSMs to gather O-D volume exists.

In Tampa and the other pilot CV test sites, researchers have been granted access to the BSMs gathered by RSUs. This information is sanitized before being released to the public, and no Personal Identifying Information (PII) is included. In addition, only the National Highway Traffic Safety Administration and motor vehicle manufacturers are able to access data directly in and identifying a particular OBU to help track potentially defective equipment. This also does not provide any PII, merely information regarding the units operation.

Many applications of CAVs do not require high-resolution data over a long stretch of roadway. For example, intersection-based applications such as red light violations only require accurate, sustained BSM vehicle ID keys in a radius around the intersection. For research purposes, depending on the traffic link that is desired to be studied, it is possible that vehicle IDs would remain constant over a sufficient length of time. This would allow individual IDs to be tracked along that period for origin-destination information. While the privacy of drivers must be taken into account, to fulfil real-time data capture individual vehicle BSMs will potentially need to be stitched together over larger distances. Researchers will need to develop algorithms or programs to account for changes in vehicle IDs if a project requires study of trajectory over a longer traffic link than trajectory is provided for in the dataset.

Rumley (32) has written a preliminary algorithm for identifying specific vehicle paths through BSM. While his algorithm produced better results than just guessing, it was still difficult to trace vehicle paths with accuracy. The USDOT acknowledges that while driver privacy through BSMs will be protected as much as possible, aftermarket GPS trackers still exist and would be a much cheaper way for someone with nefarious intent to gain vehicle path information, anyway.

SPECIFICATION 5

The BSMMS must provide for the collection of travel times.

- At the minimum, this will require algorithms that would track individual vehicle IDs, and record the following event times.
  - Vehicle crosses upstream boundary of predefined travel time section
  - Vehicle crosses downstream boundary of predefined TT section.
  - Vehicle ID appears inside a TT section without crossing the upstream boundary.
  - Vehicle ID disappears without crossing the downstream boundary.

  Tracking these events will allow for heuristic algorithms to handle the sampling of individual vehicle travel times.

Note: without lane level accuracy, collection of travel times can be very unpredictable.

SPECIFICATION 6

The BSMMS must provide Origin- Destination information

- BSMMS will require the development of specialized algorithms that allow the system to resemble the operation of a probe-based travel time and origin-destination data collection system without violating the privacy regulations required by IEEE 1609.2.
- Such algorithms must account for density on the roadway; lower density will result in an easier stitching of random vehicle paths; higher becomes more complicated
- These algorithms must account for randomly changing vehicle ID keys.

12.3.5 Collecting, Storing and Processing BSM Data

With the amount of data collected by BSMMS at any point, especially as market penetration increases, methods for storing, processing, and transferring data must be considered.

By examining an example peak hour on University Avenue in Minneapolis, MN, a sense of scale in regard to the volume of data produced by CAVs can be gained. According to the Minneapolis Traffic Counts and Crashes, on Wednesday, October 4, 2017 (representative of an average day), at University Avenue between 15th and 16th Ave SE (a high commuter block), during the AM

peak hour (7:45AM to 8:45 AM), 1,043 vehicles were counted. 100% market penetration rate is assumed for this calculation.

If a DSRC-based RSU radio has a 2000-foot diameter range (with a clear line of sight) and each vehicle is traveling at 30mph (no congestion), it would take a vehicle 44.4 seconds to pass by the radio. During the 44.4 seconds each vehicle broadcasts its BSM at 10 Hz, so 444 BSM packets are received from each vehicle. With each BSM packet being roughly 320 bytes, that means each vehicle is generating 142 kilobytes of information while in range of this particular RSU. Multiplying by the peak volume gives 148.2 megabytes of data generated at this specific 2,000ft of University Ave in just one hour.

While this may not sound like a lot of data, consider that it represents one DSRC receiver in the entire metro area. If we divide by 0.37 miles per DSRC device diameter, we get 400.5 MB per hour per mile during peak volume in the metro area. For an estimate of miles that would potentially have DSRC connections, the MnDOT snowplow records were checked (18). (Presumably, MnDOT plows all important thoroughfares.) As of 2019, 1040 miles of streets in the Minneapolis area are plowed during each snowfall. So, taking the 400.5MB per hour per mile and multiplying it by 1040 miles and 24 hours gives 9996.5 GBs per day generated in the metro area while assuming peak traffic. This is just one example case, as not all roads have volumes comparable to University Ave (much higher on freeways, or lower on less travelled arterials), and they are not all at peak capacity every hour.

If we instead assume an average of 400 vehicles per hour to account for peaks, lows, and streets with higher or lower volume than University Ave, and perform the same calculation, roughly 56.8 KB per hour per vehicle would be generated, resulting in 1,417 GB of data per day in the Minneapolis area.

Traditional loop detectors gather data at 60HZ but perform an on-the-spot aggregation into 30 second intervals, reducing the data load that is finally retained. BSMMS will need to have the same capacity for aggregation, requiring a computer in addition to the DSRC device.

SPECIFICATION 7

BSMMS must have the capacity to transmit, receive, store and aggregate the amount of data a connected roadway will produce.

12.4 Conclusion

12.4.1 General system specifications

- BSMMS must provide lane-by-lane data.
  a. BSMMS need to be able to take the information provided by the BSMs and determine which roadway and which lane of that roadway each vehicle is traveling.

- BSMMS must gather speed information as it is transmitted from vehicles.
  a. Spot-speed can be gathered from BSMs.
  b. Space-mean-speed can be calculated with an algorithm connecting the 10hz frequency transmissions along the length of the detector.

- BSMMS must provide accurate vehicle counts.
  a. At less than 100% market penetration, algorithms to infer vehicle counts based on existing BSM data must be developed.
  b. At 100% market penetration, this will be assumed.

- BSMMS must follow security protocol laid out in IEEE 1609.2.

- The BSMMS must provide for the collection of travel times.
  a. At the minimum, this will require algorithms that would track individual vehicle ID and record the times they cross predefined section boundaries or appear/disappear mid-section.
  b. Without lane level accuracy, collection of travel times can be unpredictable.

- The BSMMS must provide Origin- Destination information
  a. BSMMS must have algorithms in place to utilize a probe-based estimate of travel time and origin-destination information.
  b. These algorithms must account for density on the roadway. Lower density will result in an easier BSM path to stitch together; higher becomes more complicated.
  c. These algorithms must account for changing vehicle ID keys along a link.

- BSMMS must have the capacity to transmit, receive, store and aggregate the amount of data a connected roadway will produce.

Freeway Systems

Currently, the freeway system gathers traffic metrics every half mile. BSMMS have the potential to gather data much more frequently than that, based on 10HZz broadcast over the entire length of the DSRC range rather than just a point. In addition to the general system specifications, freeway BSMMS will be able to:

- Detect queues by:
  - Boundary density estimates
  - Vehicles slowing and stopping
- Provide data for:
  - Ramp metering
  - Travel information systems
  - Pricing
  - Incident detection

Arterial only

Traffic metrics on arterials are at present very expensive, and as a result, very spotty. Data on arterials tend to come only from actuated intersections and consists of counts and queue sizes. Having DSRC deployed throughout the city will vastly improve arterial measurements by:

- Detecting queues by:
  - Boundary density estimates
  - Vehicles slowing and stopping
  - Signalized intersection queue warning
- Providing:
  - Turning movement counts
  - Discharge and saturation flow rates
  - Limited (unknown) origin-destination information based on vehicle key change timing

13 Conclusion

In this project, the project team develops a traffic state estimation algorithm based on the Kalman filter technique and the cell transmission model.

Databases with BSMs are prepared using radar data and a microscopic simulation model in traffic simulation package AIMSUN to test the algorithm accuracy. The radar data is collected from a section on I-94. BSMs from AIMSUN focus on two freeway sections. The results show that the accuracy of the Kalman filter is good when applying to the BSMs generated by microscopic simulation models. Under the uncongested scenario, errors for the density, the speed, and the flow estimations are about 23%, 3%, and 23%, respectively. Under the congested scenario, errors for the density, the speed, and the flow estimations are about 15%, 3%, and 12%, respectively. An increase in the penetration rate increases the estimation accuracy of the Kalman filter. The accuracy of the Kalman filter is much lower when applied to BSMs from radar data as there are noises in the BSMs. The density and the flow estimations have an error of about 51%. Because the values for actual densities and flows are small, so the estimation error for these two variables is large. For speed estimation, the error is as small as 8%.

To test whether the input flow measurement at the entry of road sections can improve the accuracy, the project team adds the input flow measurement into the test database. For BSMs generated in AIMSUN, accuracies for density, speed, and flow are improved by adding input flow measurements. However, the accuracy drops for BSMs generated from radar data. One possible reason is that BSM data generated from radar data may have some errors which make the extracted values for input flow incorrect.

The project teams also created a traffic monitoring system to apply the traffic state estimation algorithm to a large network. The system is composed of data collection, data transmission, and data processing modules. In the data collection module, the system collects BSMs and uploads them to a server. BSMs collected by different road-side units are stored in different tables. In the data processing modules, local computers use BSM data on the server and estimate traffic states in real-time. Scenarios under different penetration rates are used to test the efficiency of the system. With a larger penetration rate, more BSMs need to be processed, which requires a stronger computation power. The required number of computers that guarantees real-time computation depends on the network size and the database size.

References

Connected vehicle certification for aftermarket connected vehicle world. https://www.its.dot.gov/presentations/aapex/2016_AAPEX_7layersTest.pdf.

Connected vehicle technology is coming to the streets of new york city! this technology holds the potential to make our streets safer and smarter. https://cvp.nyc/.

Connected vehicles and your privacy. https://www.its.dot.gov/factsheets/pdf/privacy_factsheet.pdf.

Cv pilot deployment program. https://www.its.dot.gov/pilots/pilots_environment.htm. Accessed: 2019-3-13.

Green hills platform for secure connected car. https://www.ghs.com/products/auto_secure_connect.html. Accessed: 2019-3-13.

How will connected vehicles be used? https://www.its.dot.gov/cv_basics/cv_basics_how_used.htm. Accessed: 2019-3-13.

Tampa cv pilot basic safety message (bsm) sample. https://data.transportation.gov/Automobiles/Tampa-CV-Pilot-Basic-Safety-Message-BSM-Sample/nm7w-nvbm. Accessed: 2019-3-26.

Thea connected vehicle pilot. https://www.tampacvpilot.com/. Accessed: 2019-3-13.

BHASKAR, A., TSUBOTA, T., CHUNG, E., ET AL. Urban traffic state estimation: Fusing point and zone based data. *Transportation Research Part C: Emerging Technologies 48* (2014), 120–142.

DENG, W., LEI, H., AND ZHOU, X. Traffic state estimation and uncertainty quantification based on heterogeneous data sources: A three detector approach. *Transportation Research Part B: Methodological 57* (2013), 132–157.

EDIE, L. C. *Discussion of traffic stream measurements and definitions*. Port of New York Authority, 1963.

FENG, Y., HOURDOS, J., AND DAVIS, G. A. Probe vehicle based real-time traffic monitoring on urban roadways. *Transportation Research Part C: Emerging Technologies 40* (2014), 160–178.

GAYAH, V. V., AND DIXIT, V. V. Using Mobile Probe Data and the Macroscopic Fundamental Diagram to Estimate Network Densities. *Transportation Research Record: Journal of the Transportation Research Board 2390*, 1 (2014), 76–86.

GORDON, R. L., AND TIGHE, W. Traffic control systems handbook (2005 edition). Tech. rep., 2005.

HARDING, J., POWELL, G., YOON, R., FIKENTSCHER, J., DOYLE, C., SADE, D., LUKUC, M., SIMONS, J., WANG, J., ET AL. Vehicle-to-vehicle communications: readiness of v2v technology for application. Tech. rep., United States. National Highway Traffic Safety Administration, 2014.

HELLINGA, B., IZADPANAH, P., TAKADA, H., AND FU, L. Decomposing travel times measured by probe-based traffic monitoring systems to individual road segments. *Transportation Research Part C: Emerging Technologies 16*, 6 (2008), 768–782.

HERRERA, J. C., AND BAYEN, A. M. Incorporation of Lagrangian measurements in freeway traffic state estimation. *Transportation Research Part B: Methodological 44*, 4 (2010), 460–481.

HERRING, R., HOFLEITNER, A., ABBEEL, P., AND BAYEN, A. Estimating arterial traffic conditions using sparse probe data. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on* (2010), IEEE, pp. 929–936.

KALMAN, R. E. A new approach to linear filtering and prediction problems. *Journal of basic Engineering 82*, 1 (1960), 35–45.

KAWASAKI, Y., HARA, Y., AND KUWAHARA, M. Traffic state estimation on a two-dimensional network by a state-space model. *Transportation Research Part C: Emerging Technologies*, March (2019), 1–17.

KHATTAK, A. J., AND WALI, B. Analysis of volatility in driving regimes extracted from basic safety messages transmitted between connected vehicles. *Transportation research part C: emerging technologies 84* (2017), 48–73.

KONG, Q.-J., LI, Z., CHEN, Y., AND LIU, Y. An approach to urban traffic state estimation by fusing multisource information. *IEEE Transactions on Intelligent Transportation Systems 10*, 3 (2009), 499–511.

LIANG, Y., CUI, Z., TIAN, Y., CHEN, H., AND WANG, Y. A deep generative adversarial architecture for network-wide spatial-temporal traffic-state estimation. *Transportation Research Record 2672*, 45 (2018), 87–105.

LIGHTHILL, M. J., AND WHITHAM, G. B. On kinematic waves. ii. a theory of traffic flow on long crowded roads. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* (1955), vol. 229, The Royal Society, pp. 317–345.

LIU, J., AND KHATTAK, A. J. Delivering improved alerts, warnings, and control assistance using basic safety messages transmitted between connected vehicles. *Transportation research part C: emerging technologies 68* (2016), 83–100.

NANTES, A., NGODUY, D., BHASKAR, A., MISKA, M., AND CHUNG, E. Real-time traffic state estimation in urban corridors from heterogeneous data. *Transportation Research Part C: Emerging Technologies 66* (2016), 99–118.

NANTHAWICHIT, C., NAKATSUJI, T., AND SUZUKI, H. Application of probe-vehicle data for real-time traffic-state estimation and short-term travel-time prediction on a freeway. *Transportation research record 1855*, 1 (2003), 49–59.

OF TRANSPORTATION, U. D. Adaptive signal control technologies: What are adaptive signal control technologies? Tech. Rep. P-41, 2018.

OFFICE, I. T. S. J. P. "knowledge resources." rita | its | costs: Unit cost components for inductive loop surveillance on corridor. Tech. rep., 2018.

PARK, Y.-K., MOON, Y.-J., CHO, Y.-S., AND KUM, K.-J. Field tests for evaluating cooperative intersection signal violation warning system (cisvws). *International Journal of Automotive Technology 14*, 2 (2013), 275–281.

RICHARDS, P. I. Shock waves on the highway. *Operations Research 4*, 1 (1956), 42–51.

RUMULY, M. C. *Privacy with DSRC Vehicle Safety Broadcasts*. PhD thesis, 2017.

SENGUPTA, R., REZAEI, S., SHLADOVER, S. E., CODY, D., DICKEY, S., AND KRISHNAN, H. Cooperative collision warning systems: Concept definition and experimental implementation. *Journal of Intelligent Transportation Systems 11*, 3 (2007), 143–155.

SEO, T., KAWASAKI, Y., KUSAKABE, T., AND ASAKURA, Y. Fundamental diagram estimation by using trajectories of probe vehicles. *Transportation Research Part B: Methodological 122* (2019), 40–56.

SEO, T., KUSAKABE, T., AND ASAKURA, Y. Traffic state estimation with the advanced probe vehicles using data assimilation. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems* (2015), IEEE, pp. 824–830.

SUN, X., MUÑOZ, L., AND HOROWITZ, R. Highway traffic state estimation using improved mixture kalman filters for effective ramp metering control. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)* (2003), vol. 6, IEEE, pp. 6333–6338.

SWINGEN, C. Improved vehicle detection and basic safety message (bsm) emulation. https://static1.squarespace.com/static/55b7b9dce4b0f8e6219738ba/t/5a053846652dea801324655e/1510291528683/4_Swingen,ISS.pdf. Accessed: 2019-3-26.

SYSTEMATICS, C. Twin cities ramp meter evaluation. Tech. rep., 2001.

TAMPÈRE, C. M., AND IMMERS, L. An extended kalman filter application for traffic state estimation using ctm with implicit mode switching and dynamic parameters. In *2007 IEEE Intelligent Transportation Systems Conference* (2007), IEEE, pp. 209–216.

TAMPÈRE, C. M., AND IMMERS, L. H. An extended Kalman filter application for traffic state estimation using CTM with implicit mode switching and dynamic parameters. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC* (2007), 209–216.

WANG, C. A Novel Approach to Estimate Freeway Traffic State : Parallel Computing and Improved Kalman Filter. 180–193.

WANG, C., RAN, B., YANG, H., ZHANG, J., AND QU, X. A novel approach to estimate freeway traffic state: Parallel computing and improved kalman filter. *IEEE Intelligent Transportation Systems Magazine 10*, 2 (2018), 180–193.

WANG, Y., AND PAPAGEORGIOU, M. Real-time freeway traffic state estimation based on extended kalman filter: a general approach. *Transportation Research Part B: Methodological 39*, 2 (2005), 141–167.

WANG, Y., PAPAGEORGIOU, M., AND MESSMER, A. Real-time freeway traffic state estimation based on extended kalman filter: A case study. *Transportation Science 41*, 2 (2007), 167–181.

WORK, D. B., TOSSAVAINEN, O.-P., BLANDIN, S., BAYEN, A. M., IWUCHUKWU, T., AND TRACTON, K. An ensemble kalman filtering approach to highway traffic estimation using gps enabled mobile devices. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on* (2008), IEEE, pp. 5062–5068.

WORK, D. B., TOSSAVAINEN, O.-P., JACOBSON, Q., AND BAYEN, A. M. Lagrangian sensing: traffic estimation with mobile devices. In *American Control Conference, 2009. ACC'09.* (2009), IEEE, pp. 1536–1543.

ZHENG, F., AND VAN ZUYLEN, H. Urban link travel time estimation based on sparse probe vehicle data. *Transportation Research Part C: Emerging Technologies 31* (2013), 145–157.

ZHU, T., KONG, X., AND LV, W. Large-scale travel time prediction for urban arterial roads based on kalman filter. In *2009 International Conference on Computational Intelligence and Software Engineering* (2009), IEEE, pp. 1–5.